



Sebastian Eichelbaum

Entwurf und Realisierung eines bidirektionalen
Datenflusses zwischen Warenwirtschaftssystem und
Web-Shop für ein Systemhaus

BACHELORARBEIT

HOCHSCHULE MITTWEIDA (FH)

UNIVERSITY OF APPLIED SCIENCES

Fachbereich Informationstechnik & Elektrotechnik

Mittweida, 2009

Sebastian Eichelbaum

Entwurf und Realisierung eines bidirektionalen
Datenflusses zwischen Warenwirtschaftssystem und
Web-Shop für ein Systemhaus

eingereicht als

BACHELORARBEIT

an der

HOCHSCHULE MITTWEIDA (FH)

UNIVERSITY OF APPLIED SCIENCES

Fachbereich Informationstechnik & Elektrotechnik

Erstprüfer: Prof. Dr.-Ing. Mario Geißler
Zweitprüfer: Dipl.-Ing. Thomas Reinhardt

vorgelegte Arbeit wurde verteidigt am:

Mittweida, 2009

Bibliografische Beschreibung:

Eichelbaum, Sebastian:

Entwurf und Realisierung eines bidirektionalen Datenflusses zwischen

Warenwirtschaftssystem und Web-Shop für ein Systemhaus. - 2009. - 88 Seiten

Mittweida, Hochschule Mittweida (FH),

Fachbereich Informationstechnik & Elektrotechnik, Bachelorarbeit, 2009

Referat:

Diese Bachelorarbeit stellt ein Konzept zur Entwicklung eines bidirektionalen Datenflusses zwischen Warenwirtschaftssystem und Web-Shop und dessen Umsetzung vor. Dabei wird die Ausgangslage analysiert und die Anforderungen an die Systeme definiert. Weiterhin wird die Umsetzung beschrieben, um den Datenaustausch und dessen Nutzung zwischen Warenwirtschaftssystem und Web-Shop zu realisieren.

Hinweis:

Alle in dieser Arbeit genannten Firmennamen, Warenzeichen, Marken und Namen von Softwareprodukten, Produkten oder Dienstleistungen sind Warenzeichen oder eingetragene Warenzeichen und damit Eigentum ihrer jeweiligen Besitzer.

Danksagung

An dieser Stelle möchte ich mich ganz herzlich bei **allen Mitarbeitern** der ProSoft Krippner GmbH Leipzig für ihre unkomplizierte und freundliche Aufnahme in Ihr Team bedanken. Bedanken möchte ich mich auch bei meinem betrieblichen Betreuer Herrn **Dipl.-Ing. Thomas Reinhardt**, der meiner Kreativität immer genug Freiraum ließ.

Ebenfalls bedanke ich mich bei meinem Betreuer an der Hochschule Mittweida Herrn **Prof. Dr.-Ing. Mario Geißler** für seine fachlichen und motivierenden Anregungen.

Nicht weniger verbunden bin ich auch meinem Arbeitskollegen **Dipl.-Inf. Stephan Pertzsch (FH)**, der mir in vielen konstruktiven Gesprächen weiterhalf.

Bedanken möchte ich mich auch bei allen **meinen Freunden**, welche ich nicht alle namentlich erwähnen kann, deren unterschiedlichste Hilfestellungen mir sehr wichtig waren.

Abschließend möchte ich noch einen besonderen Dank **meiner Mutter** aussprechen, welche mich während meiner gesamten Studienzeit immer wieder motiviert und tatkräftig unterstützt hat.

Mittweida, im März 2009

Sebastian Eichelbaum

Inhaltsverzeichnis

Abbildungsverzeichnis	iv
Tabellenverzeichnis	vi
Quelltextverzeichnis	vii
Abkürzungsverzeichnis	viii
1 Einleitung	1
1.1 Vorwort	1
1.2 Einführung	1
1.3 Zielstellung	2
1.4 Überblick	3
2 Problemanalyse	5
2.1 Ausgangslage	5
2.1.1 IT-Infrastruktur	5
2.2 Warenwirtschaft	6
2.2.1 Stammdaten und Bewegungsdaten	6
2.2.2 Stammartikel	7
2.2.3 Preislistenartikel	7
2.2.4 Distributoren	7
2.2.5 Preislisten	7
2.2.6 Mention	8
2.3 Web-Shop	8
2.4 Bestehender Datenfluss	11
2.5 Skriptsprache PHP	13
2.5.1 Objektorientierte Programmierung mit PHP	15
2.5.2 Funktionsweise einer dynamischen Webseite	15
2.5.3 Templates	16
2.5.4 PHPLIB	17
2.6 MySQL	17
2.7 UNIX	18

2.8	Zusammenfassung	19
3	Entwurf	20
3.1	Beschreibung der Funktionalitäten	20
3.1.1	Steuerung von kundenspezifischen Preisen	20
3.1.2	Steuerung und Gestaltung einer Favoritenliste	22
3.1.3	Übermittlung der Lagerbestände aus den einzelnen Filialen	23
3.1.4	Import von Vorgängen in das Warenwirtschaftssystem	24
3.2	Datenfluss	24
3.2.1	Anforderungen an den Datenfluss	25
3.2.2	Cron-Daemon	26
3.2.3	Bidirektionaler Datenaustausch	27
3.2.4	Ermittlung der benötigten Daten	28
3.2.5	Verarbeitung der ermittelten Daten	28
3.2.6	Speicherung der verarbeiteten Daten	28
3.2.7	Erstellung der Änderungsdaten	29
3.2.8	Übertragung der Änderungsdaten	30
3.2.9	Import der Änderungsdaten	30
3.2.10	Nutzung der Änderungsdaten	30
3.3	Datenbank-Design	31
3.3.1	Datenbankhilfstabellen	31
3.3.2	Datenbanktabellen	32
3.3.3	Analyse der Tabellen im Warenwirtschaftssystem	34
3.4	Integration der Daten in den Web-Shop	41
3.4.1	Abfrage der Daten	41
3.4.2	Speicherung der Daten	42
3.4.3	Aktualisierung der Daten	42
3.4.4	Anzeige der Daten	42
3.5	Zusammenfassung	43
4	Realisierung	44
4.1	Automatisierte Ausführung mit Hilfe der <i>cronjobs</i>	44
4.1.1	Anlegen der <i>cronjobs</i>	44
4.1.2	Erläuterung der <i>crontab</i> -Tabelle	46
4.2	Projektstruktur	46
4.2.1	Dateistruktur unter UNIX	47
4.2.2	Dateistruktur im Web-Shop	47
4.3	Kundenspezifische Preise	48
4.3.1	Ermittlung der Daten aus dem Warenwirtschaftssystem	48
4.3.2	Ermittlung der Änderungsdaten	49

4.3.3	Übertragung der Änderungsdaten in das Web-Shop-System	50
4.3.4	Import der Änderungsdaten in die Web-Shop-Datenbank	51
4.3.5	Anzeige der Daten im Web-Shop	51
4.4	Favoritenliste	57
4.4.1	Ermittlung der Daten aus dem Warenwirtschaftssystem	58
4.4.2	Ermittlung der Änderungsdaten	59
4.4.3	Übertragung der Änderungsdaten in das Web-Shop-System	59
4.4.4	Import der Änderungsdaten in die Web-Shop-Datenbank	60
4.4.5	Anzeige der Daten im Web-Shop	60
4.5	Übermittlung von Lagerbeständen	67
4.5.1	Ermittlung der Daten aus dem Warenwirtschaftssystem	67
4.5.2	Ermittlung der Änderungsdaten	68
4.5.3	Übertragung der Änderungsdaten in das Web-Shop-System	69
4.5.4	Import der Änderungsdaten in die Web-Shop-Datenbank	70
4.5.5	Anzeige der Daten im Web-Shop	70
4.6	Import von Vorgängen aus dem Web-Shop in das Warenwirtschafts- system	75
4.6.1	Ermittlung der Daten im Web-Shop	75
4.6.2	Aktualisierung der Daten während einer Benutzersitzung	78
4.6.3	Erstellen einer Bestellung oder eines Angebotes im Web-Shop	79
4.6.4	Übertragung der Daten in das Warenwirtschaftssystem	82
4.6.5	Import der Bestellung/Angebot in das Warenwirtschaftssystem	83
4.6.6	Angebot und Bestellung im Warenwirtschaftssystem	84
4.7	Zusammenfassung	86
5	Resümee	87
5.1	Ausblicke	87
A	Fragenkatalog	89
B	Spezifizierung der Datenbanktabellen	91
C	Klassendiagramme	94
D	Quelltexte	97
	Glossar	ix
	Literaturverzeichnis	xv
	Eidesstattliche Erklärung	xx

Abbildungsverzeichnis

2.1	IT-Infrastruktur der ProSoft Krippner GmbH	6
2.2	Das Warenwirtschaftssystem "mention"	8
2.3	Menü-Struktur im Web-Shop	9
2.4	Suchfunktionen im Web-Shop	10
2.5	Artikelaufistung im Web-Shop	10
2.6	Detailansicht im Web-Shop	11
2.7	Bestehender Datenfluss zwischen Web-Shop und Warenwirtschaftssystem	12
2.8	Architektur ab PHP4 [3]	13
2.9	Ablauf bei einer serverseitig erstellten dynamischen Webseite [10] . .	16
2.10	Übersicht über die Entwicklung der UNIX-Derviate [15]	18
3.1	Erstellen eines Angebots für kundenspezifische Preise	21
3.2	Administrationsanwendung für kundenspezifische Preise	21
3.3	Erstellen eines Angebots für eine Favoritenliste	22
3.4	Administrationsanwendung für Favoritenliste	23
3.5	Zu entwickelnder Datenfluss zwischen Web-Shop und Warenwirtschaftssystem	31
4.1	Ordnerstruktur unter UNIX	47
4.2	Ordnerstruktur im Web-Shop	47
4.3	Web-Shopartikel ohne Angebotspreis	56
4.4	Web-Shopartikel mit Angebotspreis	56
4.5	Web-Shop-Menü ohne "Mein ProSoft"	64
4.6	Web-Shop-Menü mit "Mein ProSoft"	64
4.7	"Mein ProSoft" ohne Auswahl einer Favoritenliste	64
4.8	"Mein ProSoft" mit Auswahl einer Favoritenliste	64
4.9	Favoritenliste mit Darstellung von Artikeln	65
4.10	Artikelliste mit Verfügbarkeitssymbolen	73
4.11	Lagerbestände für einen Stammartikel	74
4.12	Erläuterung zu den Verfügbarkeiten der Preislistenartikel	74
4.13	Die Warenkorbansicht im Web-Shop	77

4.14	Auswahl der Zahlungsart im Bestellvorgang	80
4.15	Abschließen des Bestellvorgangs	81
4.16	Menüpunkt im "mention" zur Umwandlung der Onlinebestellungen in interne Bestellungen	84
4.17	Onlinebestellungen vor der Umwandlung in eine interne Bestellungen	85
4.18	Onlinebestellungen nach der Umwandlung in eine interne Bestellungen	85

Tabellenverzeichnis

3.1	Datenbanktabelle <i>atAusgang</i> für die kundenspezifischen Preise	32
3.2	Datenbanktabelle <i>MentionFavoriten</i> für die Gestaltung einer Favoritenliste	33
3.3	Datenbanktabelle <i>LagerBestand</i> für die Übermittlung der Lagerbestände	33
3.4	Datenbanktabelle <i>OnlineBestellung</i> für den Import von Vorgängen (Onlinebestellung und Angebotsanfrage)	34
B.1	Datenbanktabelle " <i>Muttertabelle</i> "	91
B.2	Datenbanktabelle " <i>Mastertabelle</i> "	92
B.3	Datenbanktabelle <i>TempFavoriten</i>	93
C.1	Klassendiagramm <i>@Ausgang</i>	94
C.2	Klassendiagramm <i>Favoritenliste</i>	95
C.3	Klassendiagramm <i>Lagerbestand</i>	95
C.4	Klassendiagramm <i>Onlinebestellung</i>	96

Quelltextverzeichnis

4.1	Anzeige der <i>crontab</i> -Tabelle mit den einzelnen <i>ronjobs</i>	45
4.2	Aufruf zweier Shell-Skripte in <i>atAusgang.sh</i>	48
4.3	Aufruf eines PHP-Skriptes in <i>build_atAusgang.sh</i>	48
4.4	Log-Datei nach abarbeiten der <i>atAusgang.sh</i>	51
4.5	Überprüfung der Logindaten und der <i>@Ausgang</i> -Funktionalität	52
4.6	Auszug der <i>index.php</i> für die Funktion <i>@Ausgang</i>	52
4.7	Funktion zum Vergleich des Angebotspreises mit dem Web-Shoppreis . .	54
4.8	Ermittlung des Angebotspreises für die Anzeige in der Artikelliste	55
4.9	Initialisierung der Datenaktualisierung für <i>@Ausgang</i>	57
4.10	Überprüfung der Logindaten und der Favoritenlisten-Funktionalität . . .	60
4.11	Auszug der <i>index.php</i> für die Funktion <i>Favoritenliste</i>	61
4.12	Menücontainer in der Template-Datei <i>index.php</i> für den Menüpunkt "Mein ProSoft"	62
4.13	Ersetzen des Platzhalters für den Menüpunkt "Mein ProSoft"	63
4.14	Inhalt der Template-Datei <i>mein_prosoft.html</i> für den Menüpunkt "Mein ProSoft"	63
4.15	Erstellen einer temporären Tabelle für die Favoritenliste	65
4.16	Initialisierung der Datenaktualisierung für Favoritenliste	66
4.17	Auszug der <i>index.php</i> für die Funktion <i>Lagerbestand</i>	70
4.18	Initialisierung der Datenaktualisierung für <i>Lagerbestand</i>	75
4.19	Überprüfung der Logindaten und der Onlinebestellung-Funktionalität . .	76
4.20	Erstellen des Session-Objekts für <i>Onlinebestellung</i>	77
4.21	Löschen des Session-Objekte für <i>Onlinebestellung</i>	78
D.1	PHP-Skript für Angebote die mit " <i>@Ausgang</i> " gekennzeichnet sind . . .	97
D.2	PHP-Skript zur Ermittlung der doppelten Herstellernummern für <i>@Ausgang</i>	104
D.3	PHP-Skript zum setzten des letzten Änderungsdatum für <i>@Ausgang</i> . .	106
D.4	Shell-Skript zur Ermittlung der Änderungsdaten für <i>@Ausgang</i>	106
D.5	PHP-Skript zum Import der Änderungsdaten in den Web-Shop für <i>@Ausgang</i>	107
D.6	Auszug aus der <i>index.php</i> des Web-Shops	108

D.7 Funktion zu Ermittlung der Datenaktualität	114
D.8 Shell-Skript mit Aufruf zweier Shell-Skripte für <i>Favoritenliste</i>	114
D.9 Shell-Skript mit Aufruf eines PHP-Skripts zur Ermittlung der Daten aus dem Warenwirtschaftssystem für <i>Favoritenliste</i>	115
D.10 Shell-Skript zur Ermittlung der Änderungsdaten für <i>Favoritenliste</i>	115
D.11 PHP-Skript für Angebote die mit "langfristiges" Angebot gekennzeichnet sind	116
D.12 PHP-Skript zur Ermittlung der doppelten Herstellernummern für <i>Favo- ritenliste</i>	123
D.13 PHP-Skript zum Import der Änderungsdaten in den Web-Shop für <i>Fa- voritenliste</i>	124
D.14 PHP-Skript zur Auswahl der Ansicht einer für <i>Favoritenliste</i>	126
D.15 Shell-Skript zur Ermittlung und Übertragung der aktuellen Lagerbestände	127
D.16 PHP-Skript zur Ermittlung der Lagerbestände aus dem Warenwirtschafts- system	128
D.17 PHP-Skript zum Import der Änderungsdaten in das Intranet für Lager- bestand	130
D.18 PHP-Skript zum Import der Änderungsdaten in in den Web-Shop für Lagerbestand	131
D.19 PHP-Funktion zur Ermittlung der Symbole der Verfügbarkeiten	132
D.20 PHP-Skript für das Pop-Up-Fenstern für <i>Lagerbestand</i>	134
D.21 PHP-Funktion zur Ermittlung von Lagebestandsdaten	135
D.22 PHP-Funktion zur Ermittlung von Lagerbeständen der Filialen	135
D.23 Template-Datei <i>lager_bestand.html</i> für das Pop-Up-Fenster	135
D.24 Auszug aus dem PHP-Skript <i>warenkorb.php</i> zur Erfassung der notwen- digen Daten für Onlinebestellung	136
D.25 PHP-Skript <i>warenkorb.php</i> zum Erstellen eines Angebotes oder einer Be- stellung für die Onlinebestellung	139
D.26 Shell-Skript zur Übertragung und Import der Onlinebestellungen	144
D.27 Shell-Skript zur Übertragung der Daten für die Onlinebestellungen	144
D.28 Shell-Skript zum Import der Daten für die Onlinebestellungen	145
D.29 PHP-Skript für den Import der Daten in das Warenwirtschaftssystem	145

Abkürzungsverzeichnis

AGB	allgemeinen Geschäftsbedingungen
bzw.	beziehungsweise
ca.	circa
CSV	Comma Separated Values
evtl.	eventuell
FTP	File Transfer Protokoll
GNU	GNU is not UNIX
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protokoll
ID	Identifizier
LAMP	Linux, Apache, MySQL, PHP
MS	Microsoft
PHP	PHP Hypertext Preprocessor
scp	Secure Copy Protokoll
SQL	Structured Query Language
SSH	Secure Shell
URL	Uniform Resource Locator
VPN	Virtual Private Network
XML	Extensible Markup Language
z.B.	zum Beispiel

1 Einleitung

1.1 Vorwort

Diese Arbeit wurde im Jahr 2009 als Bachelorarbeit im Studiengang Informationstechnologie an der Hochschule Mittweida (FH) in der Firma ProSoft Krippner GmbH erstellt.

1.2 Einführung

Das World Wide Web hat sich mit der Einführung von HTTP im Jahre 1993 zu einer der wichtigsten Informationsplattformen entwickelt. Als Informationsquelle werden fachliche Foren, Online-Bibliotheken und Webseiten von renommierten Verlägen genutzt. Seit einigen Jahren gewinnen Web-Shops mehr und mehr an Bedeutung und sind zu einem festen Bestandteil des Informationsaustausches im Internet geworden.

Diese Web-Shops dienen vordergründig dem Absatz von Produkten über das Medium Internet. Darüber hinaus werden sie von vielen Benutzern genutzt, sich unkompliziert von zu Hause aus einen Überblick über das Angebot an Produkten auf dem Markt zu verschaffen. Dabei wird dem Vergleich von Preisen die größte Bedeutung zu gesprochen. Jedoch ist die Möglichkeit einer Gegenüberstellung von Produktdaten ebenso ausschlaggebend wie die Darstellung des Produktes durch ein Bild.

Möchte man als Anbieter von IT-Produkten seine Produkte zusätzlich über das Medium Internet absetzen, kann man nicht mehr auf einen Web-Shop verzichten. Dieser bietet die Möglichkeit mehr potentielle Kunden zu erreichen, als dies im normalen Tagesgeschäft möglich ist. Diesem Umstand sollte ein Web-Shop auch gerecht werden. Dazu zählt unter anderem der Umfang an Produkten, das Erscheinungsbild, die Güte der Informationen, der Umfang der bereitgestellten Funktionen und die Ergonomie.

Die hauptsächliche Nutzung des Web-Shop sollte im Vorfeld bekannt sein. Ist er, wie zum Beispiel bei den Onlinehändler Alternate oder Amazon, alleiniges Vertriebsmittel von Artikeln oder dient er vordergründig dazu, seine Kunden über neue Produkte zu

informieren oder lediglich den herkömmlichen Vertriebsweg zu unterstützen? Im zweiten Fall kann er dazu dienen, bestehende Kundenbeziehungen zu festigen oder durch seine Funktionalitäten Kunden zu begeistern und diese für den Händler zu gewinnen.

Im vorliegenden Fall wird der bestehende Web-Shop der ProSoft Krippner GmbH hauptsächlich zur Vertriebsunterstützung und Informationsplattform für deren Kunden genutzt. Daraus ist das Bestreben gewachsen, diesen Web-Shop in seinem Funktionsumfang zu erweitern und somit den Benutzern eine qualifiziertere Informationsquelle bereitstellen zu können.

1.3 Zielstellung

Ziel dieser Arbeit ist die Entwicklung und Implementierung von neuen Funktionalitäten für den Web-Shop der ProSoft Krippner GmbH. Wie im Vorfeld beschrieben dient er vorwiegend dem Informationsgewinn über IT-Produkte der ProSoft Krippner GmbH und weniger dem alleinigen Absatz von Produkten. Für die Entwicklung neuer Funktionalitäten ist es wichtig, sich über die Bedürfnisse der Benutzer im Klaren zu sein.

Demnach ergibt sich die erste Zielstellung wie folgt:

Ermittlung der zu entwickelnden Funktionalitäten durch Befragung von Kunden der ProSoft Krippner GmbH und Benutzern des bestehenden Web-Shops.

Diese Befragung wurde zu Beginn der Arbeit durchgeführt und bildet die Grundlage für die weitere Bearbeitung. Für die Befragung wurden mit ausgewählten Kunden Termine für ein persönliches Gespräch vereinbart. Diese Kunden waren für eine Befragung geeignet, da sie sich häufig in dem Web-Shop der ProSoft Krippner GmbH bewegen. Somit konnten sie eine qualifizierte Aussage über dessen Ergonomie, Funktionsumfang und eventuelle Verbesserungen geben. Für die Befragung wurde ein Fragenkatalog (siehe Anhang A auf Seite 89) entwickelt. Diese Fragen wurden in einem Interview gestellt und ließen eine freie Beantwortung durch die Befragten zu. Es stellte sich im Verlauf der Befragung heraus, dass sich die Antworten der Beteiligten sehr stark ähnelten. Dadurch konnten mit Beendigung und Auswertung dieser Umfrage konkrete Funktionalitäten ermittelt werden, welche es gilt im Folgenden umzusetzen.

Die ermittelten und umsetzbaren Funktionalitäten sind:

- Steuerung von kundenspezifischen Preisen im Web-Shop
- Steuerung und Gestaltung einer Favoritenliste im Web-Shop

- Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop
- Import von Vorgängen (Angebotsanfragen und Onlinebestellung) aus dem Web-Shop in das Warenwirtschaftssystem

Diese neuen Funktionalitäten basieren auf Daten, welche nicht im Web-Shop enthalten sind. Ferner müssen diese Daten aus dem Web-Shop in das Warenwirtschaftssystem der ProSoft Krippner GmbH importiert werden. Somit setzen diese neuen Funktionen eine Kommunikation zwischen Web-Shop und dem bestehenden Warenwirtschaftssystem voraus.

Damit ergibt sich die zweite Zielstellung wie folgt:

Entwurf und Realisierung eines bidirektionalen Datenflusses zwischen Warenwirtschaftssystem und Web-Shop.

Die Bearbeitung erfolgt durch Analyse der bestehenden Systeme Web-Shop und Warenwirtschaftssystem, sowie der Implementierung des Datenflusses und der Darstellung der übermittelten Daten.

1.4 Überblick

Die vorliegende Bachelorarbeit gliedert sich in fünf Kapitel. Im 2. Kapitel wird die Ausgangslage dieser Arbeit beschrieben. Darin werden das Warenwirtschaftssystem sowie der Web-Shop näher betrachtet. Es wird analysiert, auf welche Art und Weise die Daten des Web-Shops sowie des Warenwirtschaftssystems gehalten werden. Weiterhin werden Möglichkeiten ermittelt wie diese Daten gelesen, manipuliert und geschrieben werden können. Des Weiteren wird ein Überblick über die Technologien gegeben, die bei der Umsetzung der Aufgabenstellung zum Einsatz kommen. Dabei wird auf die Skriptsprache PHP und dem Open-Source Datenbanksystem MySQL eingegangen. Es werden hierbei die Vor- und Nachteile beider Technologien erläutert und die Aspekte hervorgehoben, die den zweckmäßigen Einsatz für dieses Projekt rechtfertigen.

Das Kapitel 3 befasst sich mit dem Entwurf des bidirektionalen Datenflusses. Darin wird die Methode erläutert, die eine "online-offline" Verbindung beider Systeme realisiert. Es wird sowohl auf die Übertragung der Daten in den Web-Shop, als auch auf den Datenimport in das Warenwirtschaftssystem eingegangen. Weiterhin wird das Design der Datenbanktabellen erläutert, die für die Datenhaltung notwendig sind.

Auf die Umsetzung des bidirektionalen Datenflusses wird im Kapitel 4 eingegangen. In diesem Kapitel wird die technische Realisierung des Datenimports bzw. Datenexports

beschrieben. Weiterhin wird erklärt, wie die notwendigen Daten ermittelt, übertragen und für die spätere Verwendung gespeichert werden.

Das letzte Kapitel fasst die Ergebnisse dieser Arbeit zusammen und liefert einen Ausblick auf die Möglichkeiten der Implementierung neuer Funktionalitäten, die auf diesem bidirektionalen Datenfluss aufbauen können.

2 Problemanalyse

Dieses Kapitel befasst sich mit den Voraussetzungen, welche den Rahmen dieser Arbeit bilden.

2.1 Ausgangslage

Die Firma ProSoft Krippner GmbH versteht sich als mittelständisches IT-Systemhaus im Raum Leipzig. Sie wurde 1990 gegründet und besitzt neben dem Hauptsitz in Delitzsch weiter Filialen in Leipzig, Dresden und Halle. Ihr Portfolio erstreckt sich über den Vertrieb von Hard- und Software, der Assemblierung einzelner PC's, der Planung, Installation und Betreuung komplexer Netzwerke bis hin zum Service und Dienstleistungen an kleinen, mittelständischen und großen Unternehmen.

ProSoft ist seit längere Zeit im World Wide Web mit Ihrer Internetpräsenz www.prosoft-krippner.com vertreten. Die Internetpräsenz umfasst die Homepage, welche dazu dient, die Firma mit deren Philosophie und Produktportfolio dem Interessierten vorzustellen. Darüber hinaus dient der firmeneigene Web-Shop der vertrieblichen Unterstützung zum Absatz von Produkten.

2.1.1 IT-Infrastruktur

Die bestehende Infrastruktur (siehe Abbildung 2.1 auf Seite 6) der Firma ProSoft Krippner gliedert sich in Intranet, Extranet und Internetzugang. Über das Extranet sind die Filialen Delitzsch und Halle über einen VPN-Tunnel an die Filiale in Leipzig angebunden. Im Intranet befinden sich der Webserver und der Datenbankserver für die Intranetseite als auch der Datenbankserver für das Warenwirtschaftssystem. Auf die Datenbestände der einzelnen Systeme kann nur intern bzw. über eine bestehende VPN-Verbindung zugegriffen werden.

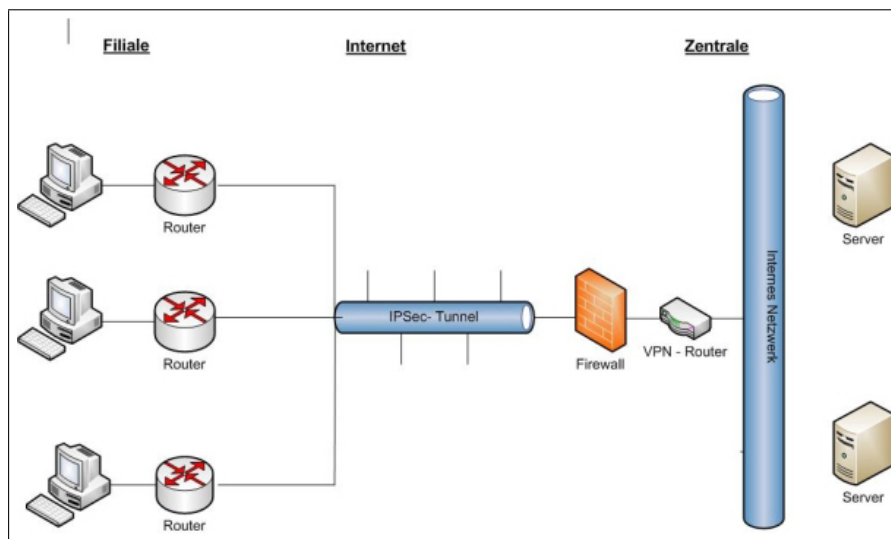


Abbildung 2.1: IT-Infrastruktur der ProSoft Krippner GmbH

Der Internetzugang wird über ein öffentliches LAMP-System realisiert. Dieses LAMP-System stellt unter anderem den Webserver und den Datenbankserver für den Web-Shop und die Homepage bereit. Die darauf enthaltenen Daten stehen über Abfragen aus dem Web-Shop bzw. der Homepage den Benutzern der Internetpräsenz zur Verfügung.

2.2 Warenwirtschaft

In diesem Abschnitt werden häufig gebrauchte Begriffe in der ProSoft Krippner GmbH näher erläutert. Dies dient dem besseren Verständnis, da diese Begriffe im weiteren Verlauf der Arbeit häufig benutzt werden.

Mit Hilfe der Warenwirtschaft werden Warenströme im Geschäftsprozess eines Unternehmens abgebildet. Dazu zählen der Einkauf und der Verkauf, die Kunden- und die Lieferantendaten, die Lagerverwaltung und die Buchhaltung.

2.2.1 Stammdaten und Bewegungsdaten

Als Stammdaten werden die Daten bezeichnet, die in der eigenen Warenwirtschaft angelegt werden. Sie zeichnen sich durch zeitliche Invarianz aus und werden langfristig gehalten. [1] Dies sind bei der ProSoft Krippner GmbH Kundendaten sowie Lieferantendaten.

Im Gegensatz zu den Stammdaten unterliegen die Bewegungsdaten einer ständigen

Änderung. Dies betrifft die Kundenaufträge und Lieferungen, die Artikel mit ihren Preisen und die Lagerbestände der vier Filialen.

2.2.2 Stammartikel

Als Stammartikel werden bei ProSoft die Artikel bezeichnet, die im Warenwirtschaftssystem angelegt sind. Durch anlegen eines Artikels bekommt dieser eine eigene interne ProSoft-Artikelnnummer.

2.2.3 Preislistenartikel

Neben den Stammartikeln gibt es noch Preislistenartikel. Damit werden die Artikel in den Preislisten der einzelnen Distributoren bezeichnet, welche es noch nicht in der eigenen Warenwirtschaft gibt. Werden diese Artikel bestellt, wandeln sich die Preislistenartikel in Stammartikel um.

2.2.4 Distributoren

Als Distributor wird im allgemeinen ein Großhändler bezeichnet. Eine Distribution beschreibt im Allgemeinen die Verteilung von Gütern von der Produktionsstätte bis zum Konsumenten. [2] Die ProSoft Krippner GmbH bezieht ihre Artikel von ca. 238 Distributoren aus ganz Deutschland. Somit ist eine große Auswahl an Artikeln sowie der günstigste Bezugspreis gegeben.

2.2.5 Preislisten

Die Preislisten werden von 25 Distributoren der Firma ProSoft Krippner bereitgestellt. Sie werden täglich bis spätestens früh um 6 Uhr per E-Mail oder FTP übermittelt. Sie dienen der Bildung der EK-Preise für das Warenwirtschaftssystem und der daraus resultierenden Anzahl an verfügbaren Artikeln im Web-Shop. Des Weiteren beinhalten die Preislisten das Zubehör zu den einzelnen Artikeln sowie die Lieferzeiten bei den Distributoren. Weiterhin sind Informationen wie die Herstellernummer, die Bezeichnung und erweiterte Benennung des Artikels, die Warengruppe und die eventuelle EOL-Kennzeichnung enthalten. Somit sind die Preislisten ein sehr wichtiges Glied in der Datenflusskette, ohne die ein tägliches arbeiten mit aktuellen Daten in dem Warenwirtschaftssystem und dem Web-Shop nicht möglich wäre.

2.2.6 Mention

Das eingesetzte Warenwirtschaftssystem in der ProSoft Krippner GmbH ist "mention" (siehe Abbildung 2.2).

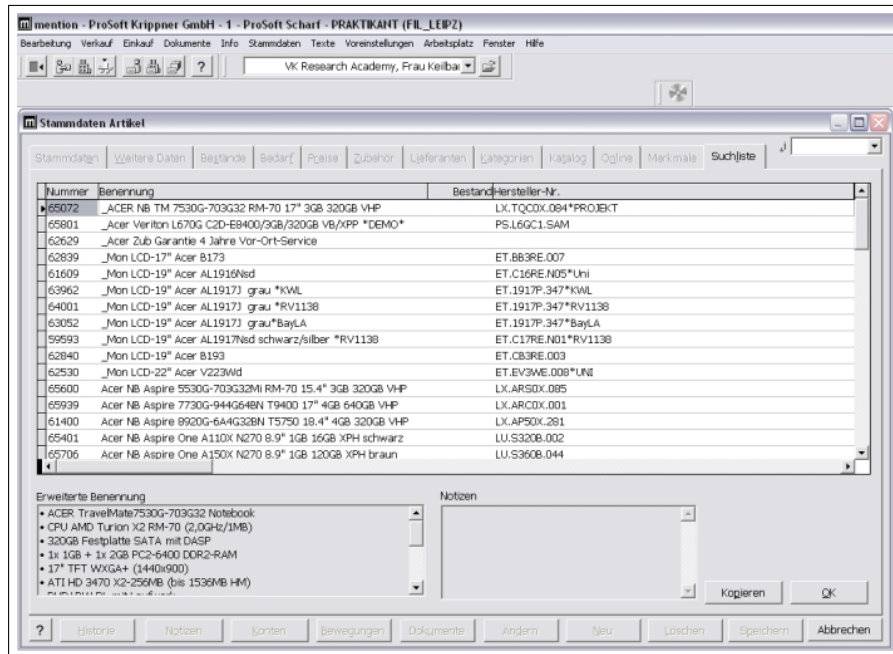


Abbildung 2.2: Das Warenwirtschaftssystem "mention"

Es wurde für mittelständische Unternehmen entwickelt und verwaltet alle betriebswirtschaftlichen Daten. Dazu zählen unter anderem die Stamm- und Bewegungsdaten. Die benötigten Daten werden zur Datenhaltung in einer Datenbank abgelegt. Die verwendete Datenbank ist eine MS-SQL-Datenbank. Auf die Datenbank kann mit Hilfe von MS SQL Server Management Studio Express zugegriffen werden. Somit können ohne Einsatz von "mention" Daten gelesen, geändert und geschrieben werden.

2.3 Web-Shop

Der bestehende Web-Shop der Firma ProSoft Krippner GmbH ist ein eigens entwickeltes Web-Content-Management-System. Damit ist es möglich, Daten aus einer Datenbank individuell zur Anzeige zu bringen. Somit können im Hintergrund Daten hinzugefügt, gelöscht oder geändert werden, ohne dass in die Programmierung des Web-Shops eingegriffen werden muss. Dieser selektiert lediglich die Daten aus der Datenbank und zeigt diese in der gewünschten Formatierung an.

Web-Content-Management-System kann mit webbasierendes Inhaltsverwaltungssystem übersetzt werden. Es zeichnet sich durch die Trennung von Anzeige der Inhalte und deren Verwaltung bzw. Pflege aus. Im vorliegenden Fall des Web-Shops sind unter Inhalte vorwiegend die Artikel mit ihren Preisen und deren Artikelbeschreibungen zu verstehen. Wie diese Artikeldaten in den Web-Shop fließen wird im Abschnitt 2.4 auf Seite 11 näher erklärt. Weiterhin können über eine Administrationsanwendung Neuigkeiten im Web-Shop, allgemeine Geschäftsbedingungen, Sonderartikel und Versandkosten von Mitarbeitern gepflegt und geändert werden.

Der Web-Shop dient der Anzeige von Artikeln mit deren Artikelbildern und Datenblättern, die die Firma ProSoft vertreibt. Darüber hinaus verfügt er über einen Warenkorb, der die darin abgelegten Artikel speichert und es somit dem Benutzer erlaubt, auf diese bei einem weiteren Besuch wieder zuzugreifen. In der Menü-Struktur werden die einzelnen Warengruppen zu logischen Einheiten zusammengefasst (siehe Abbildung 2.3).

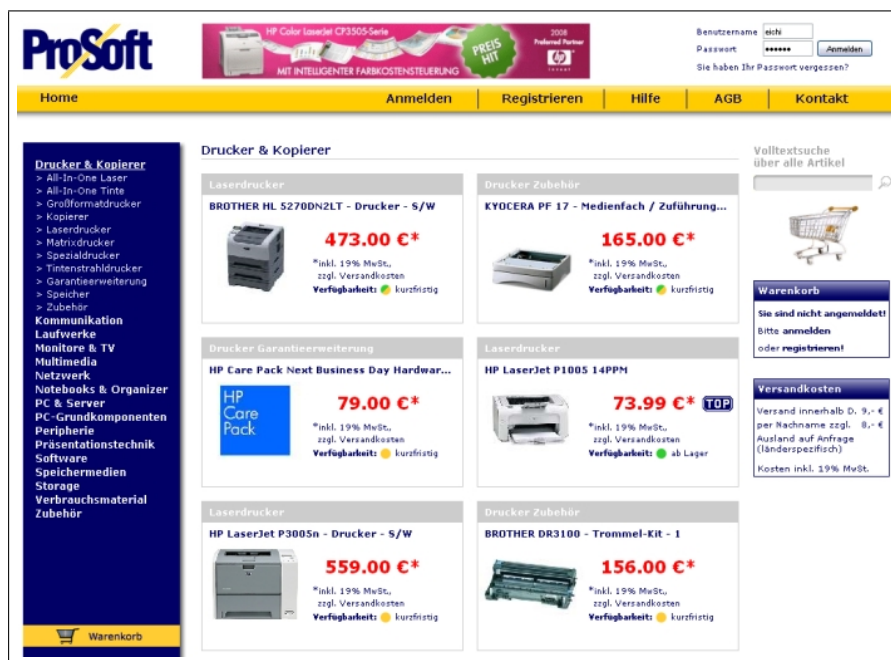


Abbildung 2.3: Menü-Struktur im Web-Shop

So findet man zum Beispiel DVD-Brenner und Festplatten unter der logischen Einheit Laufwerke, sowie Arbeitsspeicher, CPU und Mainboard unter PC-Grundkomponenten. Um gezielt Artikel im Web-Shop zu finden, verfügt dieser über zwei Suchfunktionen (siehe Abbildung 2.4 auf Seite 10).

Suche

Geben Sie hier Ihre Suchbegriffe ein (durch Leerzeichen getrennt)

Volltextsuche

Sie können Ihre Suche auch auf eine Warengruppe und/oder Hersteller beschränken

Warengruppe

Hersteller

Volltextsuche über alle Artikel

Warenkorb

Sie sind nicht angemeldet!
Bitte anmelden oder registrieren!

Versandkosten

Versand innerhalb D. 9,- €
per Nachname zzgl. 8,- €
Ausland auf Anfrage
(länderspezifisch)

Kosten inkl. 19% MwSt.

Abbildung 2.4: Suchfunktionen im Web-Shop

Zum einen ist eine Volltextsuche über alle Artikel möglich und zum anderen kann man die Suche auf Warengruppen bzw. Hersteller beschränken. Nach Anwendung einer dieser verschiedenen Suchfunktionen erhält der Benutzer eine Auflistung aller relevanten Artikel (siehe Abbildung 2.5).

All-In-One Laser

Erweiterte Suche

Hersteller: Nur ProSoft-Artikel: ☐ Info

Suche: ☐ Info Details anzeigen: ☐ Info

Suchfelder: ☐ ArtNr. ☒ Bezeichnung ☐ HerstNr. Nur Promotions: ☐ Info

Artikel pro Seite: (min. 5 max. 100)

Ihre Suchanfrage ergab 138 Treffer

ArtNr.	Bezeichnung	HerstellerNr.
50716	BROTHER DCP 7025 - MFP (s/w) - Laser	DCP7025G1
	BROTHER DCP 770CW - MFP (color) - Tintenstrahl	DCP770CWG1
	BROTHER DCP 9000 - MFP (s/w) - Laser	DCP9000G1
55630	BROTHER DCP 9000DN - MFP (s/w) - Laser	DCP9000DNG1
	BROTHER DCP 9040CN - MFP (color) - Laser	DCP9040CNG1
60949	BROTHER DCP 9045CDN - MFP (color) - Laser	DCP9045CDNG1
	BROTHER MFC 3360C - MFP (color) - Tintenstrahl	MFC-3360C01
	BROTHER MFC 5460CN - MFP (color) - Tintenstrahl	MFC5460CNG1
	BROTHER MFC 5890CN - MFP (color) - Tintenstrahl	MFC5890CN
41225	BROTHER MFC 7225n - MFP (s/w) - Laser	MFC7225NG1
41814	BROTHER MFC 8220 - MFP (s/w) - Laser	MFC8220G1
	BROTHER MFC 8460N - MFP (s/w) - Laser	MFC8460NG1
	BROTHER MFC 885CW - MFP (color) - Tintenstrahl	MFC-885CW01
47196	BROTHER MFC 8860DN - MFP (s/w) - Laser	MFC8860DNG1
	BROTHER MFC 8870DW - MFP (s/w) - Laser	MFC-8870DW

Warenkorb

Sie sind nicht angemeldet!
Bitte anmelden oder registrieren!

Versandkosten

Versand innerhalb D. 9,- €
per Nachname zzgl. 8,- €
Ausland auf Anfrage
(länderspezifisch)

Kosten inkl. 19% MwSt.

Abbildung 2.5: Artikelaufstellung im Web-Shop

Diese Liste enthält die Artikelbeschreibung, die Herstellernummer sowie die Verfügbarkeit. Preise sind erst nach einer erfolgreichen Anmeldung im Web-Shop sichtbar. Über diese Auflistung der Artikel gelangt man in die Detailansicht (siehe Abbildung 2.6

auf Seite 11) eines gewünschten Artikels.



Abbildung 2.6: Detailansicht im Web-Shop

In der Detailansicht werden das Artikelbild und das ausführliche technische Datenblatt dargestellt. Ist für diesen Artikel mindestens ein Zubehör verfügbar so wird der Reiter "Zubehör" aktiviert und der Benutzer kann sich über das verfügbare Zubehör zu diesem Artikel informieren. Die dafür benötigten Zubehördaten sind mit der Artikelnummer verknüpft und werden über die Preislisten von den Distributoren bereitgestellt.

Die Möglichkeit, einen Artikel in den Warenkorb aufzunehmen besteht sowohl in der Artikelliste als auch in der Detailansicht des Artikels.

2.4 Bestehender Datenfluss

Wie im Abschnitt 2.3 auf Seite 8 erwähnt, ist der bestehende Datenfluss für die Bildung des Web-Shops notwendig. Weiterhin ist er für die Aktualität des Warenwirtschafts-systems verantwortlich. Die Grundlage für den Datenfluss bilden die täglich von den Distributoren bereitgestellten Preislisten.

Die Informationen aus den Preislisten werden, nach dem sie an ProSoft überliefert sind, in die "Muttertabelle" importiert und gesammelt. Die Spezifizierung dieser Tabelle wird im Anhang B.1 auf Seite 91 beschrieben.

Die "Muttertabelle" wird durch Skripte in ihrem Datenbestand bereinigt. Dabei werden bestimmte Einträge verändert oder gegebenenfalls gelöscht. Zu löschen sind Daten-

sätze, die Fehler enthalten oder unerwünscht sind und somit nicht in den Web-Shop bzw. in das Warenwirtschaftssystem mit aufgenommen werden sollen. Anhand der bereinigten "Muttertabelle", welche bis zu 470.000 Datensätze beinhaltet, wird die "Mastertabelle" (siehe Anhang B.2 auf Seite 92) gebildet.

Für die Bildung der "Mastertabelle" werden Kriterien für die Benennung der Warengruppe, der Schreibweise des Herstellernamen und der Hersteller Nummer und der Bildung des EK- bzw. des VK-Preises genutzt. Somit werden die Daten der 25 Distributoren validiert und unter einem Datensatz in der "Mastertabelle" zusammengefasst. Die "Mastertabelle" beinhaltet ca. 198.000 Datensätze die für den Web-Shop relevant sind.

Die Daten aus der "Muttertabelle" werden in das Warenwirtschaftssystem importiert. In dem Warenwirtschaftssystem ist somit jeder Distributor mit seinen geliefert Daten vertreten.

Den bestehenden Datenfluss soll die Abbildung 2.7 verdeutlichen.

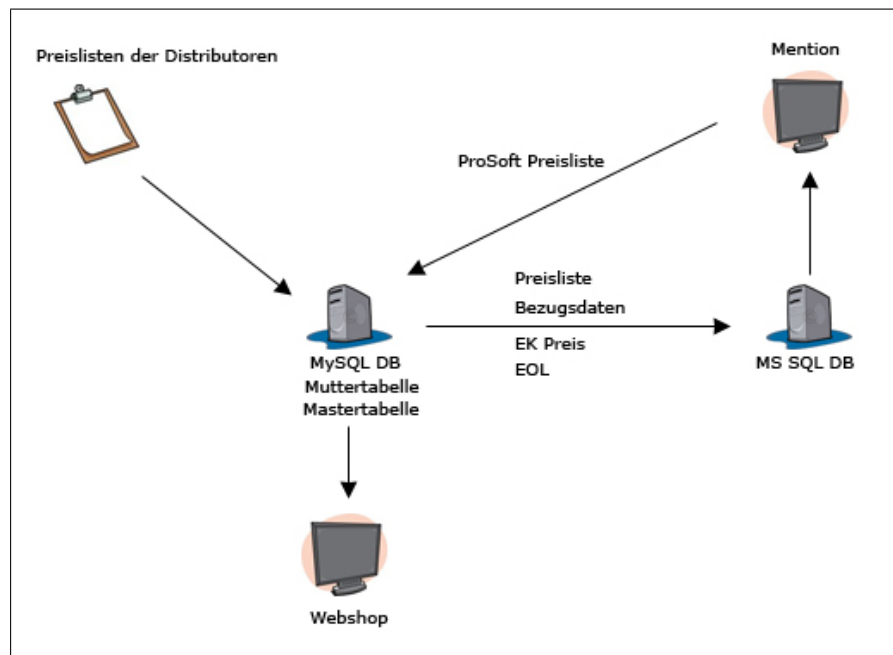


Abbildung 2.7: Bestehender Datenfluss zwischen Web-Shop und Warenwirtschaftssystem

Zusammenfassend kann gesagt werden, dass die Daten für den Web-Shop und das Warenwirtschaftssystem über den bestehenden Datenfluss bereitgestellt werden. Diese Daten sind aber von statischer Natur. Die Daten werden einmal am Tag gebildet und bleiben für die nächsten 24 Stunden unverändert. Somit können Daten, welche im Laufe des Tagesgeschäftes im Warenwirtschaftssystem geändert werden, nicht im Web-

Shop angezeigt, folglich Daten aus dem Web-Shop nicht in das Warenwirtschaftssystem importiert werden.

2.5 Skriptsprache PHP

Bezugnehmend auf die Aussage aus Abschnitt 2.3 auf Seite 8, der Web-Shop ist in Form eines Web-Content-Management-System umgesetzt, wird im Folgenden auf die verwendeten Technologien eingegangen. Die Implementierung des Web-Shops ist mit der Skriptsprache PHP in der Version PHP5 erfolgt. Ab der Version PHP4 kann der funktionelle Aufbau wie in Abbildung 2.8 dargestellt werden.

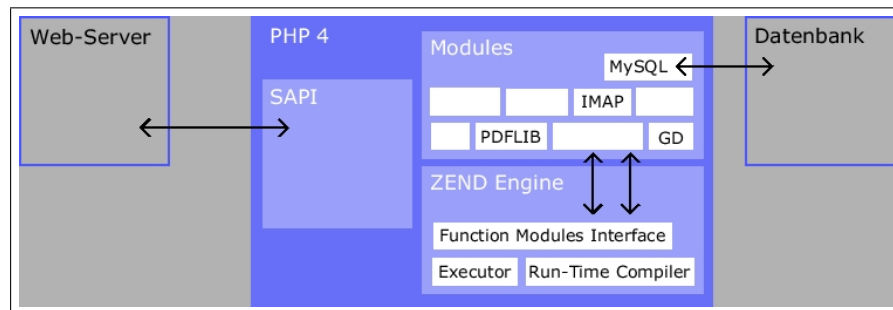


Abbildung 2.8: Architektur ab PHP4 [3]

1. ZendEngine

- der Parser der Sprache
- implementiert den Sprachkern
- analysiert den eingegebenen Quelltext, übersetzt ihn und führt diesen aus

2. Modules

- beinhaltet die einzelnen Bibliotheken
- darin ist ein Großteil der Funktionalität der Sprache implementiert
- dazu zählen z.B. die zahlreichen Schnittstellen zu Datenbanken

3. SAPI

- ist die Schnittstelle zum Webserver
- damit werden die systemunabhängigen PHP-Befehle in die serverspezifischen Befehle umgesetzt

- ein PHP-Support ist auf Grund der leicht anzupassenden SAPI nahezu für alle Webserver möglich

Mit PHP ist es möglich, dynamische Webseiten für Client-Server-Anwendungen im Internet zu erstellen. Bei PHP handelt es sich um eine Skriptsprache, die als Erweiterung in einem Webserver integriert ist. Sie unterscheidet sich von anderen Skriptsprachen, indem sie nicht wie z.B. JavaScript, auf der Clientseite im Browser des Benutzers, sondern auf dem Webserver arbeitet. Für PHP ist daher keine Unterstützung durch den Browser, sondern lediglich durch den Webserver notwendig. PHP bietet eine einfache Objektorientierung. [4]

Der Einsatz von PHP erlaubt die Verarbeitung von Formulareingaben, das Speichern und Auswerten dieser Daten, sowie die qualifizierte Rückgabe von Informationen an den Besucher der Webseite. Die Sitzungsverwaltung über Session-Variablen ist eine weitere Stärke von PHP. Damit können Besucher auf einzelnen Webseiten identifiziert und Werte von Variablen über mehrere Seiten genutzt werden.

Alle oben genannten Funktionen von PHP finden in dem Web-Shop Verwendung. Dadurch zeigt sich, dass PHP für die Umsetzung der Zielstellung geeignet und eine damit implementierte Lösung sinnvoll ist. Die Fähigkeit der objektorientierten Programmierung bestärkt zusätzlich diese Aussage. Weiterhin ist die Tatsache, dass PHP sehr gut mit dem Datenbanksystem MySQL zusammenarbeitet eine gute Voraussetzung, die Zielstellung mit dieser Technologie umzusetzen.

PHP beinhaltet als Skriptsprache einige Schwächen. Zum einen löst jeder Request an den Webserver eine Abarbeitung des PHP-Skriptes vom Anfang bis zum Ende aus. Zum anderen ist PHP zustandslos. Dies ist eine große Schwierigkeit bei der Programmierung von webbasierten Anwendungen, wie den Web-Shop der ProSoft Krippner GmbH. Zustandslos bedeutet, dass der Webserver, sobald er eine Anforderung eines Clients bearbeitet hat, die Verbindung zu dem Client wieder trennt. Alle lokalen Variablen, die in einem Skript erzeugt wurden, gehen verloren. Soll nun Interaktivität entstehen, ist es erforderlich, dass Programmzustände und Variablen gespeichert werden. Die Lösung dieses Problems ist die Implementierung eines Session-Managements. Ein Session-Management dient dazu, den bereits angemeldeten Benutzer bei der Anforderung seiner nächsten Aktion wiederzuerkennen und einen bestimmten Zustand des Skriptes wiederherzustellen. PHP ab der Version vier bietet hierfür eine geeignete Implementierung an. [5]

Diese oben genannten Erkenntnisse müssen bei der Umsetzung der Zielstellung beachtet werden.

2.5.1 Objektorientierte Programmierung mit PHP

Die objektorientierte Programmierung ist ein Konzept, mit dem Daten und Funktionen möglichst eng in einem sogenannten Objekt zusammengefasst werden. Zur besseren Verwaltung gleichartiger Objekte werden Klassen verwendet. Klassen sind Vorlagen, aus denen Objekte zur Laufzeit erzeugt werden. Klassen sind die Konstruktionspläne für Objekte. [6]

Für Dokumentation sowie für die Darstellung der Beziehung zwischen Klassen wird das Klassendiagramm verwendet. [7] Klassen werden durch Rechtecke dargestellt, die den Namen der Klasse, die Attribute und die Methoden der Klasse enthalten. Klassenname, Attribute und Methoden werden durch eine horizontale Linie getrennt. Die Attribute können durch einen Initialwert näher beschrieben werden. Sie werden aber mindestens mit ihrem Namen aufgeführt. Operationen können ebenfalls durch Parameter und Initialwerte beschrieben werden. Auch Sie werden mindestens mit ihrem Namen aufgeführt. [8]

2.5.2 Funktionsweise einer dynamischen Webseite

Serverseitig erzeugte dynamische Webseiten werden vom Webserver erst dann erstellt, wenn die Seite über einen Request vom Benutzer angefordert werden. Immer wiederkehrende Elemente sind nur ein Mal vorhanden und können zentral gepflegt werden. Somit sind Veränderungen leicht durchzuführen, wodurch sich die Wartung von Webseiten stark vereinfacht. [9]

Dynamische Webseiten zeichnen sich dadurch aus, dass

- der Inhalt der Webseite dynamisch zur Laufzeit entsteht.
- dieselbe Webseite verschiedene Inhalte darstellen kann, ohne dass sich das Layout ändert.

Der prinzipielle Ablauf bei einer serverseitig erstellten, dynamischen Webseite, kann mit Hilfe der Abbildung 2.9 aus Seite 16 erklärt werden.

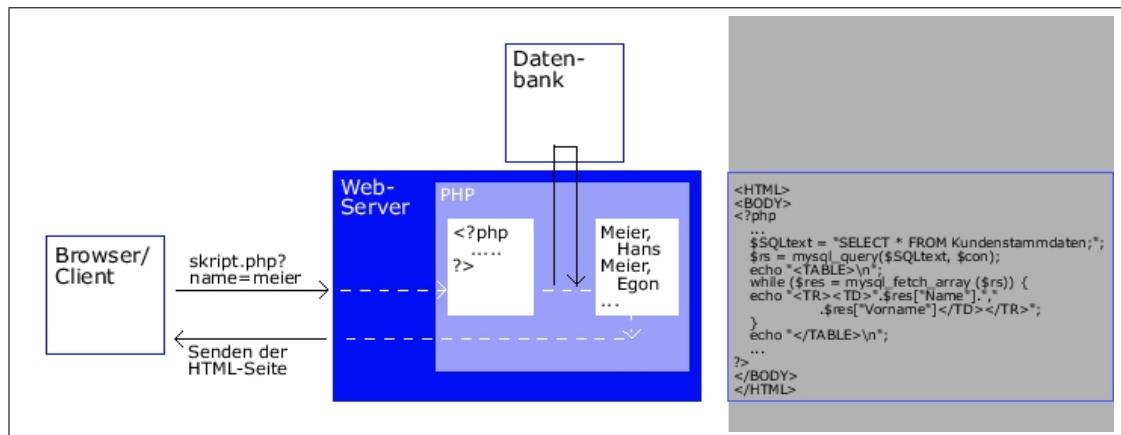


Abbildung 2.9: Ablauf bei einer serverseitig erstellten dynamischen Webseite [10]

Der Benutzer schickt über die URL im Browser einen Request an den Webserver. Durch die GET und POST Methode können beispielsweise GET- bzw. POST Variablen an den Webserver übergeben werden (`www.subdomain.de/index.php?name=meier`). Falls es sich um eine PHP-Datei handelt, wird der PHP-Parser aus der ZendEngine aufgerufen. Diesem wird die PHP-Datei zur Bearbeitung übergeben. Der PHP-Prozessor filtert die in dem Quelltext stehenden Befehle heraus und interpretiert diese. Das kann zum Beispiel eine Datenbankabfrage sein, die alle Mitglieder die den Nachnamen "Meier" haben ermittelt. Das Ergebnis dieser Datenbankabfrage wird an die jeweilige Stelle in der HTML-Datei durch den Output des PHP-Skripts gesetzt. Der Parser gibt die Seite an den Webserver zurück. Dieser liefert die fertige Webseite als Response an den Browser des Benutzers zurück.

2.5.3 Templates

Der Web-Shop basiert auf der Template-Technologie. Das Prinzip beim Erstellen von Webseiten mit Templates ist, dass Design und Skriptprogrammierung voneinander getrennt werden. Dies wird umso wichtiger, je größer der Umfang der dynamischen Webseite ist. Im Gegensatz zu statischen, rein in HTML umgesetzten Webseiten, benutzt man Platzhalter für die sich dauernd ändernden Inhalte. Die Formatierung wird in HTML und CSS geschrieben, der Inhalt wird hingegen aus einer Datenbank generiert. Das Endprodukt ist eine Vielzahl sich gleichender Webseiten mit unterschiedlichem Inhalt.

2.5.4 PHPLIB

Um die Template-Technologie nutzen zu können, ist der Einsatz von PHPLIB notwendig. PHPLIB ist eine Sammlung von in PHP geschriebenen Klassen, mit denen eine Reihe von Aufgaben gelöst werden können, die bei der Entwicklung von webbasierenden Anwendungen sehr häufig auftreten. [11]

Das in dem Web-Shop verwendete PHPLIB-Modul ist die *template.inc*, welches die Template-Engine bereitstellt. Dieses Modul wird in der *index.php* im Web-Shop durch,

```
//Einfügen der Template Klasse von Kristian Köhntopp
if(file_exists("php/template.inc")){
    include_once("php/template.inc");
}
```

über die Funktion *include_once()* inkludiert. Um die Eigenschaften dieses Modules nutzen zu können, muss eine Objektinstanz aus der Klasse *template* erzeugt werden.

```
//Erstellen einer Template - Instanz
$tpl = new template();
```

Dadurch stehen die Methoden dieser Klasse zur Verfügung und können mit *\$tpl->methodenname()* verwendet werden.

Mit diesen Methoden können unter anderem die Platzhalter in der Template-Datei durch Inhalte aus einer Datenbank ersetzt werden.

2.6 MySQL

Das zur Datenhaltung verwendete Datenbanksystem ist MySQL. Dieses System ist eines der am meist genutzten relationalen Datenbanksysteme im Open Source Bereich und bildet somit die Grundlage für viele datenbankgestützte Webauftritte. MySQL steht unter der GNU General Public License zur Verfügung. Eine relationale Datenbank speichert Daten in separaten Tabellen, statt alle Daten in einem einzigen großen Speicherraum abzulegen. Das sorgt für Vorteile hinsichtlich Geschwindigkeit und Flexibilität. [12]

Die Skriptsprache PHP bietet viele Funktionen um mit MySQL zu kommunizieren. Darunter zählen Funktionen, die mit einer MySQL-Datenbank eine Verbindung auf-

bauen oder deren Datensätze manipulieren. Damit ist es leicht, über in das PHP-Skript eingebundene SQL-Befehle, Daten aus dieser Datenbank zu lesen und zu schreiben.

Eine Administration von MySQL kann mit Hilfe von phpMyAdmin über HTTP in einem Browser erfolgen. Damit können auch Datenbanken auf fremden Rechnern über eine Netzwerkverbindung oder über das Internet administriert werden. [13]

2.7 UNIX

Bei der Umsetzung der Zielstellung kommt ein UNIX-Derivat zum Einsatz. UNIX ist ein Multitasking- und Multiuser-Betriebssystem und wurde 1971 in C geschrieben. Die Kompaktheit und strukturelle Einfachheit des Systems ermunterte viele Benutzer zur eigenen Aktivität und Weiterentwicklung des Systems. Somit erreichte UNIX recht schnell einen relativ hohen Reifegrad. Eine ähnliche Entwicklung zeigt sich seit einigen Jahren bei den freien UNIX-Derivaten zum Beispiel "Free BSD" und "Linux" für PCs. [14]

Übersicht über die Entwicklung der UNIX-Derviate

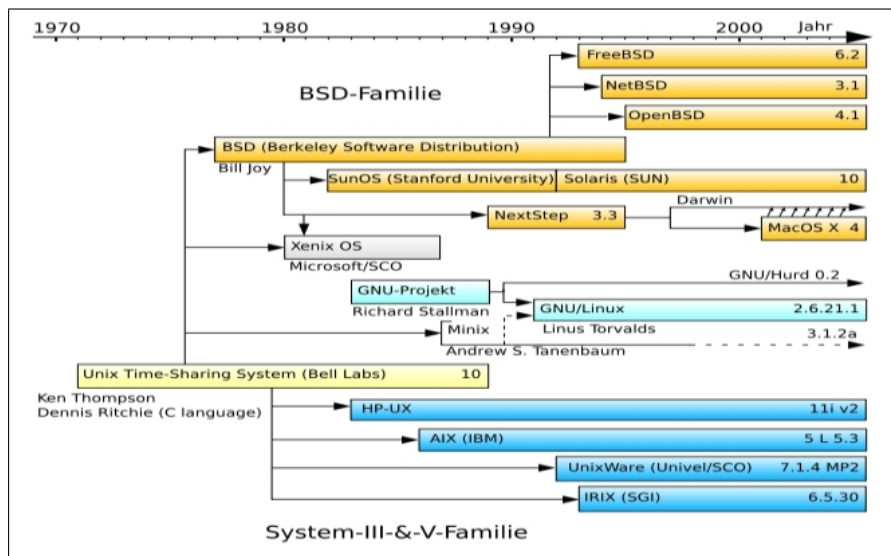


Abbildung 2.10: Übersicht über die Entwicklung der UNIX-Derviate [15]

Das bei der Umsetzung der Zielstellung eingesetzte UNIX-Derivat ist Linux in der Debian-Distribution. Debian ist ein freies Betriebssystem, basierend auf dem Linux-Betriebssystemkern. Da die meisten grundlegenden Systemwerkzeuge vom GNU-Projekt stammen, wird auch von Debian GNU/Linux gesprochen. [16]

2.8 Zusammenfassung

In diesem Kapitel wurden, neben der Beschreibung der Ausgangslage, Grundlagen für die Umsetzung der Zielstellung vermittelt. Dabei wurde auf häufig verwendete Begriffe in der Warenwirtschaft eingegangen. Weiterhin wurden Technologien erklärt, die zu den Rahmenbedingungen dieses Projektes gehören und die bei der Bearbeitung der Aufgabenstellung zum Einsatz kommen.

3 Entwurf

Dieses Kapitel beschreibt die vorgenommenen Überlegungen, um die gegebene Zielstellung zu realisieren. Dabei werden die Eigenschaften und Voraussetzungen der umzusetzenden Funktionalitäten erklärt. Des Weiteren werden Lösungsansätze diskutiert, die den Datenfluss, das Datenbank-Design und die Integration der Daten in den Web-Shop bzw. in das Warenwirtschaftssystem betreffen.

3.1 Beschreibung der Funktionalitäten

Um die gewünschten Funktionen umsetzen zu können, müssen diese in ihren Eigenschaften und Voraussetzungen analysiert werden. In diesem Abschnitt werden die einzelnen Funktionen beschrieben und Grundlagen für deren Verständnis vermittelt.

3.1.1 Steuerung von kundenspezifischen Preisen

Diese Funktion ermöglicht es, feststehende Preise im Web-Shop für ausgewählte Benutzer zu überschreiben. Dafür muss im Warenwirtschaftssystem ein Angebot für einen bestehenden Kunden erstellt werden. In dieses Angebot können mehrere Positionen eingepflegt werden. Den Preis für die jeweilige Positionen kann der Verkäufer individuell bestimmen. Damit die eingepflegten Preise für einen Kunden im Web-Shop sichtbar werden, muss das Angebot im Reiter "Detail" (schwarzer Pfeil) mit einem "Häkchen" bei "@Ausgang" (roter Pfeil) versehen werden (siehe Abbildung 3.1 auf Seite 21). Durch speichern des Angebots, wird einer eindeutigen Kundennummer (grüner Pfeil) eine eindeutige Belegnummer (blauer Pfeil) zugeordnet.

Abwicklung Verkauf - Sebastian - Angebot 2425152

Nummer 39725 Factor-Nr. Limit 0 unbegrenzt Nein 406,7526,25

Name Sebastian Eichelbaum Intern Beinhart - 0341/4531345

Strasse Leipziger Str. 147 Mit Bestellungen Typ 1 END

Ort 04552 Borna Telefon 0178,8161082

Ansprch. Fax E-Mail

Beleg A2425152

Positionen Detail Wiederrat Buchhaltung Zusatztext Beleg Detail Position

Bearbeiter 1 Reinhardt Urbeleg vom Erstellt 25.07.2008 @Eingang

Bezugnahme vom 25.07.2008 @Ausgang

ZB 1 Barzahlung ZB-Status

Anzahlung 0 Anzahlung 0,00

Versandart 0 Referenz-Nr.

Incoterms 0 GEMA 0,00

Versandanschrift 0 VK-Differenz -26,80

Rechnungsanschrift 0

Auftragsbestätigung 0

Projekt Forecast 0,00 % Direktlieferung

Lieferschein Ohne Endsummen langfristiges Angebot Ohne Preise Prior. Frachtfrei

Teillieferung zulässig Rabatt drucken automatisch Auflösen Sammelabwicklung: nur Lieferschein

Gruppen drucken mit Komponenten ohne Preise Versandkostenkontrolle nach Packen

Sammelpriorität 0 Termin

Sperre 0 Lieferung

Bank 0 Anzahl Paletten 0

Filiale FIL_LEIPZ Anzahl Pakete 0

Arbeitsplatz PRAKTIKANT 19.08.2008 13:15 Versandgewicht 0,000

Netto 551,55

Belegabatt 0,00

Versand 0,00

Verpackung 0,00

Versicherung 0,00

MwSt 19% 104,79

Brutto 656,34

Volumengewicht 11842,74 kg

Gesamtgewicht 0,00 kg ~

EAR-Gewicht 0,00 kg ~

Abbildung 3.1: Erstellen eines Angebots für kundenspezifische Preise

Die Funktion muss über eine Administrationsanwendung (siehe Abbildung 3.2) für jeden Benutzer des Web-Shops im einzelnen freigeschaltet werden (grüner Pfeil). Damit wird gewährleistet, dass die Kundennummer im Web-Shop (roter Pfeil) mit der im Warenwirtschaftssystem (schwarzer Pfeil) übereinstimmt. Die Kundennummer wird als Referenz für die Zuordnung zwischen dem Kunden und dem gespeicherten Angebot genutzt.

Kundennummer	39725
VKPreis für Kunden	
1 - Endkunde	
2 - Händler	1
6 - Prosoft-intern	
7 - BBW	
Blitzanfrage	<input checked="" type="checkbox"/>
kein "Bestellen-Button"	<input type="checkbox"/>
Alle Artikel	<input checked="" type="checkbox"/>
Suchen in MentionBezeichnung	<input checked="" type="checkbox"/>
Mention Kundennummer	39725
sieht Preise aus Belegen mit @Ausgang	<input checked="" type="checkbox"/>
sieht eine aus Mention gesteuerte Favoritenliste	<input checked="" type="checkbox"/>
kann eine OnlineBestellung in Mention durchführen	<input checked="" type="checkbox"/>

Abbildung 3.2: Administrationsanwendung für kundenspezifische Preise

Meldet sich ein Kunde im Web-Shop an, wird anhand der ermittelten Web-Shop-Kundennummer geprüft, ob ein oder mehrere Angebote mit speziellen Preisen für diese Kundennummer hinterlegt sind. Ist dies der Fall, werden die darin vorhandenen Artikelpreise eingelesen und an Stelle der allgemeingültigen Preise im Web-Shop angezeigt. Die Implementierung dieser Funktionalität wird im Abschnitt 4.3 auf Seite 48 beschrieben.

3.1.2 Steuerung und Gestaltung einer Favoritenliste

Analog der Funktion für die kundenspezifischen Preise ist es möglich, eine Favoritenliste für spezielle Benutzer des Web-Shops zu erstellen. Dafür muss ebenfalls ein Angebot für den betroffenen Kunden erstellt werden. Dieses Angebot beinhaltet Positionen, welche dem Kunden im Web-Shop als Favoriten im dafür vorgesehen Menüpunkt erscheinen.

Um diese Funktion zu aktivieren, muss im Reiter "Detail" (schwarzer Pfeil) ein "Häkchen" bei "langfristiges Angebot" (roter Pfeil) gesetzt werden (siehe Abbildung 3.3). Wird das Angebot gespeichert, ist eine Verbindung zwischen Kundennummer (grüner Pfeil) und Belegnummer (blauer Pfeil) vorhanden.

Abbildung 3.3: Erstellen eines Angebots für eine Favoritenliste

Diese Funktion (grüner Pfeil) muss ebenfalls über die Administrationsanwendung (siehe Abbildung 3.4 auf Seite 23) für jeden Kunden (roter und schwarzer Pfeil) freigeschaltet werden.




Kundennummer	39725 
VKPreis für Kunden	
1 - Endkunde	
2 - Händler	1
6 - Prosoft-intern	
7 - BBW	
Blitzanfrage	<input checked="" type="checkbox"/>
kein "Bestellen-Button"	<input type="checkbox"/>
Alle Artikel	<input checked="" type="checkbox"/>
Suchen in MentionBezeichnung	<input checked="" type="checkbox"/>
Mention Kundennummer	39725 
sieht Preise aus Belegen mit @Ausgang	<input checked="" type="checkbox"/>
sieht eine aus Mention gesteuerte Favoritenliste	<input checked="" type="checkbox"/> 
kann eine OnlineBestellung in Mention durchführen	<input checked="" type="checkbox"/>

Abbildung 3.4: Administrationsanwendung für Favoritenliste

3.1.3 Übermittlung der Lagerbestände aus den einzelnen Filialen

Diese Funktion übermittelt die aktuellen Lagerbestände aus den einzelnen Filialen in den Web-Shop. Des Weiteren kann dem Benutzer des Web-Shops eine genauere Einteilung der Artikelverfügbarkeiten angezeigt werden.

Zur Zeit verfügt jeder Artikel über drei Verfügbarkeitsstufen. Diese sind "ab Lager" (grün), "kurzfristig Verfügbar" (gelb), "auf Anfrage" (rot) und werden einmal am Tag bei der Bildung des Web-Shops aus den Preislisten bzw. dem Warenwirtschaftssystem ermittelt. Es ist nicht möglich, Bewegungen in der Warenwirtschaft im Laufe des Tages im Web-Shop darzustellen. Ebenfalls ist eine Differenzierung auf eine bestimmte Filiale, in der sich ein Artikel auf Lager befindet, nicht gegeben.

Die aktuellen Lagerbestände können nur für Stammartikel aus dem Warenwirtschaftssystem ermittelt werden. Die Verfügbarkeiten der Preislistenartikel werden weiterhin aus den Preislisten entnommen. Jedoch wird die Verfügbarkeit "kurzfristig Verfügbar" genauer spezifiziert. Ein Artikel der sich bei mindestens einem Distributor auf Lager befindet, erhält die Verfügbarkeit "verfügbar Lieferant" (grün-gelb) und ein Artikel mit einer Lieferzeit bei einem Distributor von kleiner 10 Werktagen die Verfügbarkeit "kurzfristig Verfügbar" (gelb).

Erhält ein Artikel den Status "ab Lager" (grün), wird dem Benutzer des Web-Shops angezeigt, in welcher Filiale sich dieser Artikel auf Lager befindet.

3.1.4 Import von Vorgängen in das Warenwirtschaftssystem

Die letzte umzusetzende Funktion ist der Import von Onlinebestellungen bzw. Angebotsanfragen in das Warenwirtschaftssystem. Diese Funktion soll den Arbeitsablauf der Mitarbeiter verbessern und vereinfachen.

Zur Zeit werden Bestellungen oder Angebotsanfragen aus dem Web-Shop per E-Mail an einen Mitarbeiter der Firma ProSoft übermittelt. Darin ist der Kunde mit seinen persönlichen Daten und die Bestellpositionen mit Artikel- und Herstellernummer, Preisen und Mengenangaben enthalten. Anhand dieser Daten muss der zuständige Mitarbeiter die Bestellung in das Warenwirtschaftssystem eintragen. Diesen Umstand soll diese Funktion verbessern.

Bestellungen und Angebotsanfragen werden automatisch in das Warenwirtschaftssystem importiert. Dabei werden alle erforderlichen Daten, wie Kundennummer, Artikel- und Herstellernummer, EK- bzw. VK-Preis und die Mengenangabe in das System eingetragen. Der zuständige Mitarbeiter erhält lediglich eine E-Mail zur Information, dass eine neue Bestellung in das Warenwirtschaftssystem eingegangen ist. Diese importierte Onlinebestellung muss in eine interne Bestellung umgewandelt werden. Dies ist im Warenwirtschaftssystem, durch bereits vorhandene Funktionen, sehr schnell möglich.

3.2 Datenfluss

In diesem Abschnitt wird auf den Entwurf des bidirektionalen Datenflusses eingegangen. Dieser soll den automatischen Datenaustausch zwischen Warenwirtschaftssystem und Web-Shop realisieren.

Wie im Abschnitt 2.4 auf Seite 11 beschrieben, bietet der bestehende Datenfluss keine Möglichkeit Daten zwischen Warenwirtschaftssystem und Web-Shop auszutauschen. Dies basiert auf der Tatsache, dass der Datenbestand für das Warenwirtschaftssystem und den Web-Shop einmal am Tag gebildet und dieser in die jeweiligen Systeme importiert wird. Eine Übermittlung von Änderungen der Daten zwischen beiden Systemen ist nicht möglich. Ziel des neu zu entwerfenden Datenflusses ist es, die Übertragung von Änderungen aus einem System in das andere zu ermöglichen.

3.2.1 Anforderungen an den Datenfluss

Der Datenaustausch soll folgende grundlegende Eigenschaften besitzen:

- automatisiert
- zeitlich gesteuert
- bidirektional
- indirekte Verbindung
- Änderungsübermittlung

Der Datenabgleich soll selbsttätig ablaufen. Damit ist eine menschliche Initialisierung bzw. Steuerung nicht notwendig. Zu welcher Zeit der Austausch der Daten stattfinden soll, wird ebenfalls zentral geregelt. Somit stehen die neuen Daten in festgelegten Zeitabständen zur Verfügung.

Damit beide Systeme die Daten erhalten bzw. diese an das andere System weitergeben können, ist die Übertragung der Daten in beide Richtungen notwendig. Das bedeutet, dass sowohl Daten aus den Web-Shop in das Warenwirtschaftssystem als auch Daten aus dem Warenwirtschaftssystem in den Web-Shop übermittelt werden müssen. Ein direkter Bezug der Daten aus dem jeweiligen System ist nicht erwünscht. Dies hat sicherheitsrelevante Aspekte. Wäre dies möglich, könnte der Web-Shop dazu benutzt werden, Daten direkt im Warenwirtschaftssystem zu ändern oder zu löschen.

Damit bei dem ständig wiederkehrenden Datenfluss die Infrastruktur des Netzwerkes nicht unzweckmäßig belastet wird, ist von einer Übermittlung des gesamten Datenbestandes abzuraten. Vielmehr sind lediglich die Änderungen der Daten von Interesse. Dadurch wird das zu übertragende Datenvolumen auf ein Minimum reduziert und die betroffenen Datenbanken entlastet. Somit hat der Datenfluss weniger negative Auswirkungen auf die zur Verfügung stehende Bandbreite im Intranet.

Des Weiteren ist eine höhere Verfügbarkeit des Web-Shops sowie des Warenwirtschaftssystems gegeben. Bei dem Import der Daten in eine Datenbank ist diese für den Zeitpunkt des Imports gesperrt. In dieser Zeit können keine Requests bearbeitet werden und der Web-Shop bzw. das Warenwirtschaftssystem steht nicht zur Verfügung.

Die geforderten Eigenschaften können wie folgt umgesetzt werden:

- automatisierter und zeitlich gesteuerter Datenabgleich mit *cronjobs* unter Linux
- bidirektionaler Datenaustausch durch Übertragung der Daten in beide Richtungen

- eine indirekte Verbindung durch zwischenspeichern der Daten in Hilfstabellen
- Übermittlung von Änderungen im Datenbestand durch erstellen von Differenzdateien mit *diff* unter Linux

3.2.2 Cron-Daemon

Linux besitzt mit dem *cron*-Deamon eine geeignete Möglichkeit, um den automatisierten zeitlich wiederkehrenden Datenaustausch zu initialisieren.

Der cron-Daemon ist eine Jobsteuerung von UNIX bzw. UNIX-Derivaten die wiederkehrende Aufgaben (*cronjobs*) automatisch zu einer bestimmten Zeit ausführen kann. Häufig führt der cron-Daemon wichtige Programme für die Instandhaltung des Systems aus, wie zum Beispiel Dienste für das regelmäßige Archivieren von Logdateien. [17]

Mit Hilfe dieser Jobsteuerung kann eine automatische, wiederkehrende Initialisierung des Datenaustausches erfolgen. Dafür sind im Vorfeld sinnvolle Zeitabstände des Datenabgleiches festzulegen. Für die Ermittlung der zeitlichen Abfolge kann man die anfallenden Abläufe in zwei Kategorien einteilen. Zum einen in die zeitkritischen und zum anderen in die zeitunkritischen Abläufe. Bei den zeitkritischen Abläufen bedarf es einem kurzen Intervall des Datenabgleiches, um Änderungen im Warenwirtschaftssystem bzw. im Web-Shop schnellstmöglich dem anderen System zur Verfügung zu stellen. Der Abgleich des Datenbestandes für die zeitunkritischen Abläufe kann in größeren zeitlichen Abständen erfolgen.

Unter diesen Kategorien werden die im Abschnitt 1.3 auf Seite 2 herausgearbeiteten und umzusetzenden Funktionen wie folgt eingeordnet:

zeitkritische Abläufe (Datenabgleichsintervall kleiner, gleich 10 Minuten)

- Steuerung von kundenspezifischen Preisen im Web-Shop
- Steuerung und Gestaltung einer Favoritenliste im Web-Shop
- Import von Vorgängen (Angebot und Bestellung) aus dem Web-Shop in das Warenwirtschaftssystem

zeitunkritische Abläufe (Datenabgleichsintervall größer, gleich 1 Stunde)

- Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop

Diese Einteilung wird getroffen, da Änderungen bei den zeitkritischen Abläufen während einer Benutzersitzung wahrscheinlicher sind als bei den zeitunkritischen Abläufen.

Ein Benutzer kann sich längere Zeit in dem Web-Shop bewegen. Dies kann auch während eines Telefonats mit seinem zugeteilten Betreuer oder einem Mitarbeiter der Firma ProSoft geschehen. Werden innerhalb eines Telefonats spezielle Konditionen zu Preisen oder eine Favoritenliste aus dem Warenwirtschaftssystem für diesen Kunden und Benutzer des Web-Shop erstellt, müssen diese Daten während einer Sitzung des Benutzers in den Web-Shop übermittelt werden.

Um eine automatische Steuerung der Abläufe zu implementieren, müssen die *cronjobs* unter Linux angelegt werden. Die Implementierung der *cronjobs* mittels *crontab* unter Linux wird im Abschnitt 4.1 auf Seite 44 beschrieben.

3.2.3 Bidirektionaler Datenaustausch

Der bidirektionale Datenaustausch kann in zwei Kategorien zusammengefasst werden. Zum einen werden Daten in das Warenwirtschaftssystem und zum anderen Daten in den Web-Shop importiert. Die im Abschnitt 1.3 auf Seite 2 herausgearbeiteten und umzusetzenden Funktionen werden wie folgt in diese Kategorien eingeordnet:

Übertragung von Daten in das Warenwirtschaftssystem

- Import von Vorgängen (Angebot und Bestellung) aus dem Web-Shop in das Warenwirtschaftssystem

Übertragung von Daten in den Web-Shop

- Steuerung von kundenspezifischen Preisen im Web-Shop
- Steuerung und Gestaltung einer Favoritenliste im Web-Shop
- Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop

3.2.4 Ermittlung der benötigten Daten

Zu Beginn des Datenflusses ist es notwendig, die benötigten Daten aus dem Web-Shop bzw. dem Warenwirtschaftssystem zu beziehen. Die Bezugsquelle der Daten aus dem Web-Shop ist eine MySQL-Datenbank und die Bezugsquelle der Daten aus dem Warenwirtschaftssystem ist eine MS-SQL-Datenbank. Wie im Abschnitt 2.1.1 auf Seite 5 beschrieben, befinden sich die Bezugsquelle des Web-Shops im Internet und die Bezugsquelle des Warenwirtschaftssystem im Intranet. Beide Bezugsquellen sind logisch von einander getrennt. Somit kann man die Daten nur in dem System ermitteln, in dem sie auch gehalten werden. Daten aus dem Web-Shop können nur in diesem System ermittelt, und müssen während einer Sitzung des Benutzers im Web-Shop gesammelt werden. Sie sind für die Umsetzung der Importvorgänge (Onlinebestellung und Angebotsanfrage) aus dem Web-Shop in das Warenwirtschaftssystem notwendig. Die Daten aus dem Warenwirtschaftssystem können nur im Intranet ermittelt werden. Diese Daten sind für die Umsetzung der Steuerung von kundenspezifischen Preisen und Gestaltung einer Favoritenliste im Web-Shop, sowie der Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop notwendig.

3.2.5 Verarbeitung der ermittelten Daten

Damit die Daten des jeweiligen Systems verwendet werden können, müssen diese nach ihrer Ermittlung verarbeitet werden. Die Verarbeitung der Daten erfolgt ebenfalls in dem System, in dem sie ermittelt werden. Die Verarbeitung der Daten im Web-Shop muss während einer Sitzung des Benutzers im Web-Shop erfolgen. Dazu ist die Implementierung einer geeigneten Routine im Web-Shop erforderlich. Die Daten aus dem Warenwirtschaftssystem müssen ebenfalls über geeignete, zu implementierende Routinen verarbeitet werden.

3.2.6 Speicherung der verarbeiteten Daten

Die ermittelten und verarbeiteten Daten aus beiden Systemen können, wie im Abschnitt 3.2.4 aus Seite 28 erwähnt, nicht direkt in das jeweils andere System importiert werden. Daraus ergibt sich die Notwendigkeit der Datenspeicherung. Die Speicherung der verarbeiteten Daten aus dem Web-Shop oder dem Warenwirtschaftssystem erfolgt in Datenbankhilfstabellen. Diese Hilfstabellen müssen im Vorfeld mit geeigneten Feldern entworfen werden (siehe Abschnitt 3.3 auf Seite 31).

Die Hilfstabelle zur Speicherung von Daten aus dem Web-Shop beinhaltet die benötigten Daten für den Import von Vorgängen (Onlinebestellung und Angebotsanfrage) aus dem Web-Shop in das Warenwirtschaftssystem. Diese Datenbankhilfstabelle wird in der MySQL-Datenbank des LAMP-Systems erstellt. Auf die Anforderungen an diese Datenbankhilfstabelle wird in Abschnitt 3.3.2 auf Seite 33 eingegangen. Die Hilfstabellen zur Speicherung von Daten aus dem Warenwirtschaftssystem beinhalten die benötigten Daten zur Steuerung von kundenspezifischen Preisen und zur Gestaltung einer Favoritenliste im Web-Shop, sowie der Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop. Diese Datenbankhilfstabellen werden in der MySQL-Datenbank des Intranets angelegt. Auf die Erstellung dieser Datenbankhilfstabellen wird im Abschnitt 3.3.2 auf Seite 32 und im Abschnitt 3.3.2 auf Seite 33 näher eingegangen.

3.2.7 Erstellung der Änderungsdaten

Wie im Abschnitt 3.2.1 auf Seite 25 beschrieben, ist eine Übertragung von Änderungsdaten sinnvoll. Die Änderung zwischen dem bestehenden und dem neuen Datenbestand muss im Vorfeld ermittelt werden. Dafür bietet Linux eine geeignete Möglichkeit. Diese besteht darin, mit dem Befehl *diff* zwei Textdateien miteinander zu vergleichen und den Unterschied zwischen beiden Dateien in eine neuen "Differenz-Datei" zu schreiben. Diese beinhaltet nur die Änderungsdaten und kann nachfolgend für den Import der Änderung in das jeweilige System verwendet werden.

Die Änderungsdaten sind nur für die Übertragung von Daten in den Web-Shop notwendig. Dies begründet sich darin, dass der Import von Vorgängen (Onlinebestellung und Angebotsanfrage) aus dem Web-Shop in das Warenwirtschaftssystem nicht von statischer Natur ist. Onlinebestellung und Angebotsanfrage, welche erfolgreich in das Warenwirtschaftssystem importiert wurden, werden sowohl in der Datenbank des Web-Shops als auch in der des Intranets gelöscht. Somit tritt in jedem Fall eine Änderung der Daten in den Abgleichsintervallen auf. In den Fällen der kundenspezifischen Preise, der Favoritenlisten bzw. der Lagerbestände muss dies nicht immer auftreten.

Die Implementierung zur Ermittlung der Änderungsdaten wird im Abschnitt 4.3.2 auf Seite 49 bzw. im Abschnitt 4.5.2 auf Seite 68 erläutert.

3.2.8 Übertragung der Änderungsdaten

Wie im Abschnitt 3.2.7 auf Seite 29 erwähnt, müssen nur Änderungsdaten, welche für einen Import in den Web-Shop notwendig sind anhand der vorher erstellten "Differenz-Dateien" aus dem Intranet in das Web-Shop-System übertragen werden. Dies geschieht per *scp* über eine mit SSH gesicherte Verbindung. Ist die Übertragung abgeschlossen, kann mit Hilfe dieser "Differenz-Dateien" der Import der Änderungsdaten in den Web-Shop erfolgen.

Wie die Daten für den Import von Vorgängen (Onlinebestellung und Angebotsanfrage) aus dem Web-Shop in das Warenwirtschaftssystem übertragen und importiert werden, wird im Abschnitt 4.6.4 auf Seite 82 näher erklärt.

3.2.9 Import der Änderungsdaten

Sind die Änderungsdaten in das Web-Shop-System übertragen, müssen anhand dieser die entsprechenden Datenbanktabellen verändert werden. Die Änderung der Datensätze kann durch löschen, verändern oder hinzufügen neuer Daten erfolgen. Dazu dienen die übertragenen "Differenz-Dateien". Diese werden eingelesen und daraus die zu löschen, zu verändernden und hinzuzufügenden Datensätze ermittelt. Die Implementierung dieses Imports ist im Abschnitt 4.3.4 auf Seite 51 bzw. im Abschnitt 4.5.4 auf Seite 70 nachzulesen.

3.2.10 Nutzung der Änderungsdaten

Nach der Übertragung und des Imports der Daten in den Web-Shop bzw. in das Warenwirtschaftssystem können diese genutzt werden. Dazu muss der Web-Shop diese Daten abfragen, verarbeiten und in geeigneter Form darstellen (siehe Abschnitt 4.3.5 auf Seite 51 bzw. Abschnitt 4.5.5 auf Seite 70).

Das Warenwirtschaftssystem kann selbständig mit diesen Daten arbeiten, wenn diese vorher in festgelegten Tabellen und Spalten importiert werden.

Zu entwickelnder Datenfluss zwischen Web-Shop und Warenwirtschaftssystem

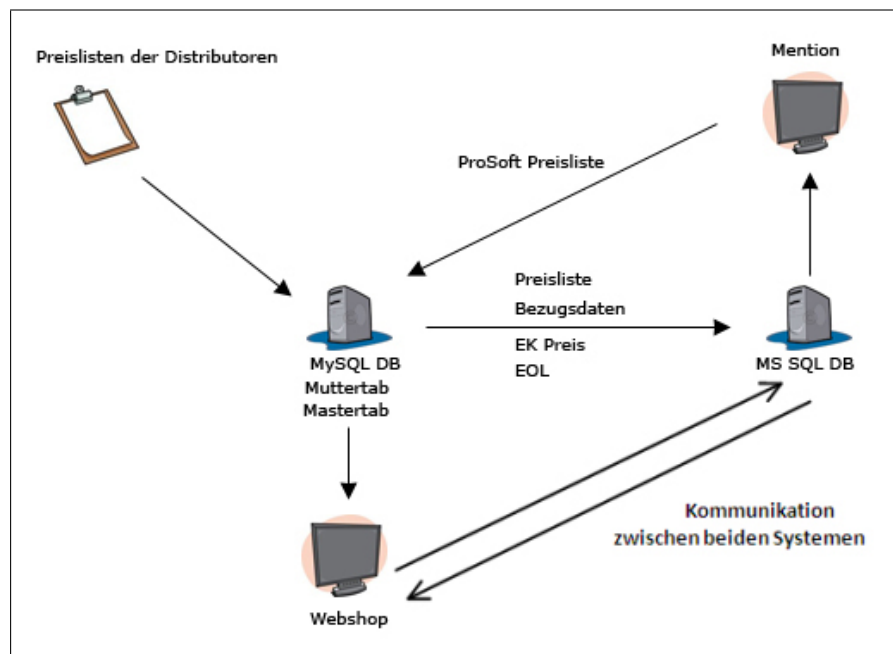


Abbildung 3.5: Zu entwickelnder Datenfluss zwischen Web-Shop und Warenwirtschaftssystem

3.3 Datenbank-Design

Dieser Abschnitt befasst sich mit dem Entwurf der Datenbanktabellen. Dies umfasst die benötigten Hilfstabellen als auch die Datenbanktabellen für die Datenhaltung. Die Datenhaltung im Warenwirtschaftssystem basiert auf Datenbanktabellen, welche schon im System enthalten sind.

3.3.1 Datenbankhilfstabellen

Wie im Abschnitt 3.2.6 auf Seite 28 erwähnt, dienen die Hilfstabellen der temporären Speicherung der ermittelten Daten aus dem Web-Shop und dem Warenwirtschaftssystem. Sie realisieren damit den indirekten Datenaustausch. Im Verlauf der Datenermittlung werden diese Tabellen genutzt, um nur relevante Datensätze für einen Differenzvergleich zu erhalten. Um einen Vergleich zwischen Hilfstabellen und Tabellen aus dem Web-Shop realisieren zu können, müssen beide Tabellen jeweils die gleichen Spalten besitzen. Hilfstabellen werden für die folgenden Funktionalitäten benötigt:

- Steuerung von kundenspezifischen Preisen im Web-Shop

- Steuerung und Gestaltung einer Favoritenliste im Web-Shop
- Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop
- Import von Vorgängen (Angebot und Bestellung) aus dem Web-Shop in das Warenwirtschaftssystem

3.3.2 Datenbanktabellen

Um die übermittelten Daten speichern, und für den Web-Shop bzw. dem Warenwirtschaftssystem bereit stellen zu können, bedarf es Tabellen in den jeweiligen Datenbanken. Diese Tabellen müssen alle relevanten Informationen zur Umsetzung der Funktionalitäten speichern können.

Datenbanktabellen *atAusgang* und *MentionFavoriten*

Diese Datenbanktabellen sind für die Funktionalität der Steuerung von kundenspezifischen Preisen und Gestaltung einer Favoritenliste im Web-Shop notwendig. Diese beide Funktionen können zusammengefasst werden, da sie wie im Abschnitt 3.1.2 auf Seite 22 beschrieben, analog erstellt werden und das gleiche Design der Datenbanktabellen voraussetzen. Bei der Tabelle *atAusgang* kommen lediglich Informationen zum Netto- und Bruttopreis hinzu.

Das Design der Datenbanktabelle *atAusgang* für die Funktion der kundenspezifischen Preise soll die Tabelle 3.1 verdeutlichen:

Tabelle 3.1: Datenbanktabelle *atAusgang* für die kundenspezifischen Preise

Spaltenname	Beschreibung	Datentyp	Länge
bsbelnr	Belegnummer	VARCHAR	20
bskundennr	Kundennummer	VARCHAR	20
hnr	Herstellernummer	VARCHAR	50
bpvkpreis	Netto VK-Preis	FLOAT	10,2
bpvkbrutt	Brutto VK-Preis	FLOAT	10,2
last_mod	letzte Änderung des Belegs (UNIX-Timestamp)	VARCHAR	10
last_update	letztes Update der Tabelle	DATETIME	

Das Design der Datenbanktabelle *MentionFavoriten* für die Funktion der Gestaltung einer Favoritenliste soll die Tabelle 3.2 verdeutlichen:

Tabelle 3.2: Datenbanktabelle *MentionFavoriten* für die Gestaltung einer Favoritenliste

Spaltenname	Beschreibung	Datentyp	Länge
bsbelnr	Belegnummer	VARCHAR	20
bskundennr	Kundennummer	VARCHAR	20
hnr	Herstellernummer	VARCHAR	50
last_mod	letzte Änderung des Belegs (UNIX-Timestamp)	VARCHAR	10
last_update	letztes Update der Tabelle	DATETIME	

Datenbanktabelle *LagerBestand*

Diese Datenbanktabelle ist für die Funktionalität der Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop notwendig. Das Design der Datenbanktabelle *LagerBestand* soll die Tabelle 3.3 verdeutlichen:

Tabelle 3.3: Datenbanktabelle *LagerBestand* für die Übermittlung der Lagerbestände

Spaltenname	Beschreibung	Datentyp	Länge
ArtNr	ProSoft-Artikelnummer	INT	11
hnr	Herstellernummer	VARCHAR	50
Delitzsch	Bestand in Filiale Delitzsch	INT	11
Halle	Bestand in Filiale Halle	INT	11
Hauptlager	Bestand in Filiale Leipzig	INT	11
Lieferbkt	Art der Verfügbarkeiten	ENUM	
last_update	letztes Update der Tabelle	DATETIME	

Tabelle *OnlineBestellung*

Diese Datenbanktabelle ist für die Funktionalität des Imports von Vorgängen (Onlinebestellung und Angebotsanfrage) aus dem Web-Shop in das Warenwirtschaftssystem notwendig. Das Design der Datenbanktabelle *OnlineBestellung* soll die Tabelle 3.4 auf Seite 34 verdeutlichen:

Tabelle 3.4: Datenbanktabelle *OnlineBestellung* für den Import von Vorgängen (Onlinebestellung und Angebotsanfrage)

Spaltenname	Beschreibung	Datentyp	Länge
bestellid	generierte eindeutige Bestellungs-ID	VARCHAR	20
oskundennr	Kundennummer aus dem Web-Shop	INT	11
hnr	Herstellernummer	VARCHAR	50
aridnr	interne Artikelnummer des Warenwirtschaftssystems für Versandkosten und Nachnahmegebühr (nur gefüllt bei Versandkosten)	VARCHAR	10
osbeltyp	Belegtyp (Angebot "A" oder Bestellung "B")	VARCHAR	1
oszeit	Uhrzeit der Bestellung	VARCHAR	6
osdatum	Datum der Bestellung	DATETIME	
oszusatz	Zusatztext zu der Bestellung	TEXT	
opvkpreis	Nettopreis der Position	FLOAT	15,2
opstueck	Mengenangabe der Position	FLOAT	15,3
opvkbrutt	Bruttopreis der Position	FLOAT	15,2
Name	Nachname des Bestellers	TINYTEXT	

3.3.3 Analyse der Tabellen im Warenwirtschaftssystem

Um Daten aus dem Warenwirtschaftssystem zu ermitteln bzw. Daten in dieses zu importieren, ist es notwendig, das vorhandene Design der zugrunde liegenden Datenbank und deren Tabellen zu analysieren. Dabei kann bei der Analyse die folgende Einteilung getroffen werden.

Analyse der Tabellen im Warenwirtschaftssystem für:

- Steuerung von kundenspezifischen Preisen im Web-Shop
- Steuerung und Gestaltung einer Favoritenliste im Web-Shop
- Übermittlung der Lagerbestände aus den einzelnen Filialen in den Web-Shop
- Import von Vorgängen (Angebot und Bestellung) aus dem Web-Shop in das Warenwirtschaftssystem

Analyse der Datenbanktabellen für die kundenspezifischen Preise und der Gestaltung einer Favoritenliste

Bei der Analyse der im Warenwirtschaftssystem eingesetzten Datenbanktabellen kann für diese beiden Funktionen ebenfalls eine Analogie angenommen werden. Dies beruht auf der Tatsache, dass beide Funktionen auf dem gleichen Datenbestand basieren. Einziger Unterschied sind die gesetzten "Häkchen" in den Angeboten ("@Ausgang" oder "langfristiges Angebot").

Die zu betrachtenden Tabellen sind:

- BESTKK
- BESTKP
- AEL
- PRL

Die Datenbanktabelle BESTKK

Diese Tabelle kann mit "Bestellung Kunden Kopf" übersetzt werden. Sie beinhaltet alle Kopfdaten einer Bestellung bzw. eines Angebots. Die Kopfdaten sind allgemeine Daten, die eine Bestellung oder ein Angebot beschreiben. Dies betrifft unter anderem Daten über das Erstelldatum, den Belegtyp, die Belegnummer und die Kundennummer.

Die für die Funktionen benötigten Kopfdaten sind:

- BSBELNR (Belegnummer, Verknüpfung zu BESTKP)
- BSKUNDENNR (Kundennummer)
- BSWSZEIT (Uhrzeit der Belegerstellung)
- BSDATUM (Datum der Belegerstellung)
- BSBELTYP (Belegtyp, "A" für Angebot, "B" für Bestellung)
- BSEMENAU (gesetztes "Häkchen" bei "@Ausgang")
- BSLANGFR (gesetztes "Häkchen" bei "langfristiges Angebot")
- BSMANKEY (Mandant)

Die Datenbanktabelle BESTKP

Diese Tabelle kann mit "Bestellung Kunden Positionen" übersetzt werden. Sie beinhaltet alle Positionsdaten einer Bestellung bzw. eines Angebots. Die Positionsdaten beschreiben die Eigenschaften der Position, welche in einer Bestellung oder eines Angebots vorhanden ist. Dies betrifft unter anderem Daten über den Netto- und Bruttopreis, die Stückzahl der Position, den Distributor für die Position und die Bestellnummer der Position bei einem Distributor.

Die für die Funktionen benötigten Positionsdaten sind:

- BPIDNR (interne Datensatznummer der Position, gefüllt bei Stammartikel, Verknüpfung zu AEL)
- BPDISTRIB (Distributor der Position, gefüllt bei Preislistenartikel, Verknüpfung zu PRL)
- BPBESTNR (Bestellnummer der Position beim Distributor, gefüllt bei Preislistenartikel, Verknüpfung zu PRL)
- BPVKPREIS (Nettopreis der Position)
- BPVKBRUTT (Bruttopreis der Position)
- BPBELNR (Belegnummer der Position, Verknüpfung zu BESTKK)

Die Datenbanktabelle AEL

Diese Tabelle kann mit "Stammartikel" übersetzt werden. Sie beinhaltet alle Daten der Stammartikel im Warenwirtschaftssystem. Dies sind unter anderem die Herstellernummer und die Artikelbenennung eines Stammartikels.

Die für die Funktionen benötigten Stammmartikeldaten sind:

- ARHERSTNR (Herstellernummer des Artikels)
- ARIDNR (interne Datensatznummer, Verknüpfung zu BESTKP)

Die Datenbanktabelle PRL

Diese Tabelle kann mit "Preislistenartikel" übersetzt werden. Sie beinhaltet alle Daten der Preislistenartikel im Warenwirtschaftssystem. Dies sind die Herstellernummer, die Artikelbenennung, der Distributor und die Bestellnummer bei einem Distributor

Die für die Funktionen benötigten Preislistenartikeldaten sind:

- PRHERSTNR (Herstellernummer der Position)
- PRDISTRIB (Distributor der Position, Verknüpfung zu BESTKP)
- PRBESTNR (Bestellnummer bei dem Distributor, Verknüpfung zu BESTKP)

Analyse der Datenbanktabellen für die Übermittlung der Lagerbestände

Um die Lagerbestände der einzelnen Filialen aus dem Warenwirtschaftssystem ermitteln zu können, muss die zugrunde liegende Datenbanktabelle analysiert werden.

Die zu analysierende Datenbanktabelle ist:

- AELLAGER

Die Datenbanktabelle AELLAGER

Diese Tabelle kann mit "Stammartikel Lager" übersetzt werden. Sie beinhaltet alle Lagerdaten der Stammartikel im Warenwirtschaftssystem. Dies sind die Standorte der Lager wie zum Beispiel das Hauptlager, das Lager in Delitzsch und das Lager in Halle und der Lagerbestand eines jeweiligen Artikels.

Die für die Funktion benötigten Lagerdaten sind:

- MLLAGER (Name des Lagers (Hauptlager, Delitzsch, Halle))
- MLBESTAND (Lagerbestand des Artikels)
- MLIDNR (interne Datensatznummer der Position, gefüllt bei Stammartikel, Verknüpfung zu AEL)
- MLMANKEY (Mandant)

Analyse der Datenbanktabellen für den Import von Vorgängen in das Warenwirtschaftssystem

Damit die Onlinebestellungen und Angebotsanfragen in das Warenwirtschaftssystem importiert werden können, müssen alle Datenbanktabellen analysiert werden, die diese Funktion realisieren.

Die folgenden Datenbanktabellen sind für diese Funktion notwendig und müssen näher betrachtet werden:

- OBESTKK
- OBESTKP
- KUNDEN
- ADRESSPA

Weiterhin sind die Tabellen AEL und PRL für den Import in das Warenwirtschaftssystem notwendig.

Die Datenbanktabelle OBESTKK

Diese Tabelle kann mit "Online Bestellung Kunden Kopf" übersetzt werden. Sie benötigt alle Kopfdaten einer zu importierenden Onlinebestellung bzw. Angebotsanfrage. Dies sind unter anderem die Kundennummer, der Belegtyp, der gesamte Netto- bzw. Bruttopreis der Bestellung, der Zusatztext der Bestellung, das Bestelldatum und die Uhrzeit der Bestellung.

Die für die Funktion benötigten Positionsdaten sind:

- OSMANKEY (Mandant)
- OSKUNDENNR (Kundennummer)
- OSZB (Zahlungsbedingung des Kunden)
- OSZEIT (Uhrzeit der Bestellung)
- OSDATUM (Datum der Bestellung)
- OSWAEHRUNG (Währung, festgelegt auf Euro)

- OSUKURS (Umrechnungskurs, festgelegt auf 1)
- OSUEINH (Umrechnungseinheit, festgelegt auf 1)
- OSUDATUM (Erstelldatum der Bestellung)
- OSTEXT (Text für die Bestellung (Ihre Onlinebestellung vom))
- OSBESTDAT (Datum der Bestellung)
- OSZUSATZT (Zusatztext der Bestellung)
- OSPALFDNR (Ansprechpartner des Kunden, Verknüpfung zu ADRESSPA)
- OSBRUTTO (gesamter Bruttopreis der Bestellung)
- OSMWST (Mehrwertsteuersatz der Bestellung, festgelegt auf 1)
- OSVKPREIS (gesamter Nettopreis der Bestellung)
- OSBELTYP (Belegtyp ("A" für Angebot, "B", für Bestellung))

Die Datenbanktabelle OBESTKP

Diese Tabelle kann mit "Online Bestellung Kunden Position" übersetzt werden. Sie benötigt alle Positionsdaten einer zu importierenden Onlinebestellung bzw. Angebotsanfrage. Dies sind unter anderem Netto- und Bruttopreis, die Stückzahl der Position, den Distributor für die Position und die Bestellnummer der Position bei einem Distributor.

Die für die Funktion benötigten Positionsdaten sind:

- OPMANKEY (Mandant)
- OPLFDNUM (laufende Nummer)
- OPIDNR (interne Artikelnummer der Position, gefüllt bei Stammartikel, Verknüpfung zu AEL)
- OPVKPREIS (Nettopreis der Position)
- OPSTUECK (Stückzahl der Position)
- OPMWST (Mehrwertsteuer, festgelegt auf 1)
- OPWAEHRUNG (Währung, festgelegt auf Euro)

- OPUKURS (Umrechnungskurs, festgelegt auf 1)
- OPUEINH (Umrechnungseinheit, festgelegt auf 1)
- OPVKBRUTT (Bruttopreis der Position)
- OPDISTRIB (Distributor der Position, gefüllt bei Preislistenartikel, Verknüpfung zu PRL)
- OPBESTNR (Bestellnummer der Position beim Distributor, gefüllt bei Preislistenartikel, Verknüpfung zu PRL)

Die Datenbanktabelle KUNDEN

Diese Tabelle kann mit "Kundendaten" übersetzt werden. Sie beinhaltet alle Daten eines Kunden der im Warenwirtschaftssystem angelegt ist.

Die für die Funktion benötigten Kundendaten sind:

- KDZB01 (Zahlungsbedingungen des Kunden)
- KDNUMMER (Kundennummer)
- KDADNUMMER (Verknüpfung zu ADRESSPA)

Die Datenbanktabelle ADRESSPA

Diese Tabelle kann mit "Adressdaten Ansprechpartner" übersetzt werden. Sie beinhaltet alle Daten über die Ansprechpartner eines Kunden, der im Warenwirtschaftssystem angelegt ist.

Die für die Funktion benötigten Adressdaten der Ansprechpartner sind:

- PALFDNR (laufende Nummer des Ansprechpartners)
- PANAME2 (Nachname des Kunden)
- PANUMMER (Verknüpfung zu KUNDEN)

3.4 Integration der Daten in den Web-Shop

Dieser Abschnitt befasst sich mit den Lösungsansätzen, wie die ermittelten und übertragenen Daten in den Web-Shop integriert werden können. Dabei werden Gesichtspunkte wie Performance, Ergonomie und die Abfrage und Speicherung der Daten besprochen.

Die Performance einer Anwendung ist ein wichtiger Gesichtspunkte, welcher bei der Implementierung beachtet werden muss. Sie ist ausschlaggebend für die Akzeptanz einer Anwendung durch den Benutzer. Lange Wartezeiten auf angeforderte Daten und langsames reagieren der gesamten Anwendung sind von einem Benutzer nicht erwünscht. Somit ist es um so wichtiger, sich im Vorfeld der Implementierung über eine performante Umsetzung der Zielstellung Gedanken zu machen.

Die Performance in diesem Projekt kann in folgenden Bereichen betrachtet und analysiert werden:

- Abfrage der Daten
- Speicherung der Daten
- Aktualisierung der Daten
- Anzeige der Daten

3.4.1 Abfrage der Daten

Damit die relevanten Daten für den Web-Shop bereit gestellt werden können, müssen diese aus der Web-Shop-Datenbank abgefragt werden. Die Abfrage der Daten erfolgt mittels SQL-Befehlen. Um die benötigten Daten zu erhalten sollten möglichst wenige "Selects" erforderlich sein. Dies entlastet den Datenbankserver und hält die Antwortzeiten des Web-Shops auf Anfragen des Benutzers so niedrig wie möglich. Weiterhin ist die Tatsache zu beachten, dass sich die abgefragten Daten nur in den vorher festgelegten Intervallen ändern können. Somit ist eine einzige Abfrage der Daten bei Beginn der Benutzersitzung im Web-Shop ausreichend. Die Abfrage der Daten muss innerhalb der Erneuerungszeiträume wiederholt werden, da sich, wie im Abschnitt 3.2.2 auf Seite 26 erwähnt, die Daten im Warenwirtschaftssystem und in der Web-Shop-Datenbank während einer Benutzersitzung verändern können und deshalb neu eingelesen werden müssen. Dadurch lassen sich die Abfragen auf ein Minimum reduzieren und eine Entlastung des Datenbankservers ist gegeben.

3.4.2 Speicherung der Daten

Damit dem Benutzer in einer Sitzung die abgefragten Daten zur Verfügung stehen, müssen diese temporär gespeichert werden. Zur Speicherung der Daten bieten sich Session-Objekte an. Das Session-Management wird durch PHP bereitgestellt. Damit ist es möglich, die ermittelten und für eine Sitzung des Benutzers relevanten Daten temporär zu speichern. Die benötigten Daten können in einem Session-Array abgelegt werden. Die darin enthaltenen Daten werden durch PHP auf dem Webserver gespeichert. Diese Daten werden mit einer dem Benutzer zugeordneten Session-ID verknüpft. Diese Session-ID wird wiederum als Cookie im Browser des Benutzers gespeichert. Somit stehen alle Daten aus dem Session-Array zur Verfügung, ohne das jedesmal auf die Datenbank zugegriffen werden muss.

3.4.3 Aktualisierung der Daten

Befindet sich ein Benutzer länger als das Erneuerungsintervall im Web-Shop, so müssen die in dem Session-Array gespeicherten Daten aktualisiert werden. Durch initialisieren einer neuen Datenbankabfrage können die neuesten Daten in das Session-Objekt gespeichert werden. Den Zeitpunkt für die Initialisierung muss eine geeignete Methode ermitteln.

3.4.4 Anzeige der Daten

Wichtig für die Akzeptanz der neuen Funktionen ist die Darstellung der gewonnenen Daten im Web-Shop. Um die Daten übersichtlich und nutzerfreundlich anzeigen zu können, ist der Entwurf eines Layouts sinnvoll. Des Weiteren sollten die neuen Funktionen schnell über geeignete Menüpunkte zu erreichen sein.

Die Ansicht der Favoritenliste ist unter dem Menüpunkt "Mein ProSoft" zu erreichen und lehnt sich an die Darstellung der Artikelliste an. Somit ist ein Wiedererkennungseffekt in Optik und Bedienung der Favoritenliste im Vergleich zur Artikelliste vorhanden.

Die Kennzeichnung der kundenspezifischen Angebotspreise erfolgt in der Artikelliste durch ein geeignetes Symbol.

Die Anzeige der Lagerbestände wird in das bestehende Anzeigesystem der Verfügbarkeiten integriert. Somit treten für den Benutzer des Web-Shops für diese Funktion keine Änderung in der Bedienung auf.

3.5 Zusammenfassung

In diesem Kapitel wurden Lösungsansätze für die Umsetzung der Zielstellung gegeben und diese diskutiert. Ziel war die Vermittlung von Grundlagen, die für das weitere Verständnis bei der Realisierung der Aufgabenstellung notwendig sind. Dabei wurden die umzusetzenden Funktionen erläutert und der Datenfluss, das Datenbankdesign und die Integration der Daten in den Web-Shop näher beschrieben.

Es wurde dabei festgestellt, dass die im Web-Shop verwendeten Technologien PHP und MySQL für die Umsetzung der Aufgabenstellung geeignet sind. Sie besitzen Eigenschaften mit denen es möglich ist, die geforderten Funktionen unter Berücksichtigung von Performance und Ergonomie umzusetzen.

Die bestehenden Datenbanken im Intranet sowie im Web-Shop-System können so modifiziert bzw. erweitert werden, dass ein indirekter bidirektionaler Datenfluss zwischen Web-Shop und Warenwirtschaftssystem realisiert werden kann.

4 Realisierung

In diesem Kapitel wird die praktische Umsetzung der gestellten Aufgaben beschrieben.

4.1 Automatisierte Ausführung mit Hilfe der *cronjobs*

Wie im Abschnitt 3.2.2 auf Seite 26 beschrieben, kann die automatisierte Ausführung des bidirektionalen Datenabgleichs zwischen Web-Shop und Warenwirtschaftssystem mit Hilfe von *cronjobs* erfolgen. Diese *cronjobs* werden von dem UNIX-Dienst *cron*-Deamon bereitgestellt. Dieser Abschnitt beschreibt, wie die *cronjobs* erstellt werden und den Datenfluss für die einzelnen Funktionen initialisieren.

4.1.1 Anlegen der *cronjobs*

Damit eine wiederkehrende Initialisierung des Datenflusses erfolgen kann, müssen die dazu notwendigen Befehle in einer Tabelle aufgeführt werden. Diese Tabelle wird *crontab* genannt. Sie besteht aus sechs Spalten; die ersten fünf dienen der Zeitangabe (Minute, Stunde, Tag, Monat, Wochentag), alle weiteren Zeichen bis zum Zeilenumbruch werden als der auszuführende Befehl aufgefasst. Die einzelnen Spalten werden durch Leerzeichen oder Tabulatoren getrennt. [18]

Damit der *crontab* eingerichtet werden kann muss der Befehl

```
crontab -e
```

auf der UNIX-Kommandozeile aufgerufen werden. Der übergebene Parameter *-e* erlaubt die *crontab*-Datei zu bearbeiten bzw. zu erstellen, falls diese noch nicht existiert. Der Quelltext 4.1 auf Seite 45 zeigt die erstellte *crontab*-Datei mit den darin aufgelisteten *cronjobs* zur Initialisierung des Datenflusses für die umzusetzenden Funktionen.

```

1 #
2 #crontab for pl-import
3 #
4 #
5 SHELL=/bin/sh
6 PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin
7 HOME=/var/log
8 #
9 #minute hour mday month wday command
10 #
11 #erstellt die Kunden, KundenBetreuer und pers-Tabelle auf Nexus-neu (um mit der
    MentionKundennummer des Bestellers die Betreuer EMail zu ermitteln um EOL-
    Artikel per Mail zu versenden)
12 */10 0-5,7-23 * * * /usr/local/pl-update/bin/Mention2Shop/shell/importBestellung
    /create_Kunden_KundenBetreuer.sh > /usr/local/pl-update/bin/Mention2Shop/log
    /shell/create_Kunden_KundenBetreuer.log 2>&1
13 #
14 #erstellt die atAusgang-Tabelle auf NEXUS-NEU mit den PHP-Dateien (
    select_beleg_aushgang.php, clear_hnr.php)
15 #anschliessend kopiert die atAusgang-Tabelle auf TRINITY.atAusgang
16 */10 0-5,7-23 * * * /usr/local/pl-update/bin/Mention2Shop/shell/atAusgang/
    atAusgang.sh > /usr/local/pl-update/bin/Mention2Shop/log/shell/atAusgang.log
    2>&1
17 #
18 #erstellt die MentionFavoriten-Tabelle auf NEXUS-NEU mit den PHP-Dateien (
    select_liste_favoriten.php, clear_hnr.php)
19 #anschliessend kopiert die MentionFavoriten-Tabelle auf TRINITY.MentionFavoriten
20 */10 0-5,7-23 * * * /usr/local/pl-update/bin/Mention2Shop/shell/listeFavoriten/
    listeFavoriten.sh > /usr/local/pl-update/bin/Mention2Shop/log/shell/
    listeFavoriten.log 2>&1
21 #
22 #kopiert die Tabelle onlineBestellung von Trinity auf Nexus-Neu
23 #anschliessend importiert die Bestellungen ins Mention OBESTKK und OBESTKP mit
    der PHP-Datei (get_online_bestellung.php)
24 */10 0-5,7-23 * * * /usr/local/pl-update/bin/Mention2Shop/shell/importBestellung
    /importBestellung.sh > /usr/local/pl-update/bin/Mention2Shop/log/shell/
    importBestellung.log 2>&1
25 #
26 #aktualisieren der Lagerbestaende auf Nexus und Trinity mit den Lagerbestaenden
    aus Mention
27 0 0-5,9-23 * * * /usr/local/pl-update/bin/Mention2Shop/shell/bestandLager/
    bestandLager.sh > /usr/local/pl-update/bin/Mention2Shop/log/shell/
    bestandLager.log 2>&1

```

Quelltext 4.1: Anzeige der *crontab*-Tabelle mit den einzelnen *rognjobs*

4.1.2 Erläuterung der *crontab*-Tabelle

Die oben aufgeführte *crontab*-Tabelle enthält vier Einträge. Diese Einträge dienen der Initialisierung des Datenabgleichs der einzelnen Funktionen. Alle Einträge können analog betrachtet werden, da sich die Vorgehensweise beim Erstellen eines Eintrags in die *crontab*-Tabelle gleicht.

Der erste Eintrag ist für die Funktion der kundenspezifischen Preise im Web-Shop notwendig. Die erste Spalte **/10* erlaubt ein ausführen des *cronjobs* aller zehn Minuten. Die zweite Spalte *"0-5,7-23"* dient der Eingrenzung der Stunden, in denen eine Ausführung stattfinden soll. Der Datenfluss bei dieser Funktion wird in der sechsten Stunde eines jeden Tages unterbrochen. Zu diesem Zeitpunkt findet die Bildung des Web-Shop aus der Mutter- und Mastertabelle, wie im Abschnitt 2.4 auf Seite 11 beschrieben, statt. Somit ist ein Datenabgleich nicht notwendig und kann die Bildung des Web-Shops negativ beeinflussen. Die dritte, vierte und fünfte Spalte sind mit einem *** versehen, da sie in diesen Fällen keine Anwendung finden. Sie dienen der Eingrenzung der Ausführung auf einen Tag, einen Monat und einer Woche im Jahr.

Die Initialisierung des Datenabgleichs erfolgt mit dem Befehl in der sechsten Spalte. Dieser ruft das auszuführende Programm oder eine Datei auf. In diesem Fall ist dies ein Shell-Script, dessen Ausgabe mit *>* in eine Log-Datei geschrieben wird. Mit der Anweisung *2>&1* wird eine evtl. auftretende Fehlermeldung des *crontab* nach */dev/null* umgeleitet.

Sind alle *cronjobs* in die *crontab*-Tabelle eingetragen, kann diese Datei mit *:wq* gespeichert und beendet werden. Somit werden alle zeitgesteuerten Befehle dem *cron*-Deamon bekannt gegeben und mit der Kommandozeilenausgabe

```
crontab: installing new crontab
```

bestätigt.

4.2 Projektstruktur

Um die Aufgabenstellung realisieren zu können, müssen diverse Dateien angelegt werden. Dies umfasst die Shell- und PHP-Skripte als auch Konfigurationsdateien. Dieser Abschnitt beschreibt die Dateistruktur, welche diesem Projekt zugrunde liegt.

4.2.1 Dateistruktur unter UNIX

Alle in dem Projekt zu erstellenden Dateien, werden in dem Hauptpfad *usr/local/bin/Mention2Shop* im Intranet abgelegt. Somit ist eine strikte Trennung zu anderen Projekten gegeben. In diesem Ordner findet eine Trennung zwischen *Log*-, *Shell*- und *PHP*-Dateien statt. In diesen Unterordnern werden weiterhin die einzelnen Themengebiete gegliedert. Darin befinden sich die einzelnen Dateien, die für eine Implementierung erstellt werden. Die Ordnerstruktur kann, ausgehend von */usr/local/bin/Mention2Shop*, wie folgt dargestellt werden:



Abbildung 4.1: Ordnerstruktur unter UNIX

4.2.2 Dateistruktur im Web-Shop

Die Struktur der Ordner und Dateien im Web-Shop ist eine typische Webstruktur. Sie gliedert sich in Ordner für die HTML-Dateien, den PHP-Skripten, den Bildern und den Formatvorlagen. Die Ordnerstruktur kann, ausgehend von */www/htdocs/*, wie folgt dargestellt werden:



Abbildung 4.2: Ordnerstruktur im Web-Shop

4.3 Kundenspezifische Preise

Dieser Abschnitt beschreibt die Umsetzung zur Anzeige von kundenspezifischen Preisen im Web-Shop der Firma ProSoft Krippner GmbH.

4.3.1 Ermittlung der Daten aus dem Warenwirtschaftssystem

Damit individuelle Preise für Produkte angezeigt werden können, müssen diese Preise aus den erstellten Angeboten im Warenwirtschaftssystem ausgelesen werden. Da die Datenhaltung des Warenwirtschaftssystems im Intranet erfolgt, können die Preise nicht direkt aus dem Warenwirtschaftssystem ermittelt werden. Auf das Warenwirtschaftssystem kann nur aus dem Intranet zugegriffen werden. Der Zugriff erfolgt über in PHP geschriebene Selects. Die Selects ermitteln die notwendigen Daten aus der MS-SQL-Datenbank, die dem Warenwirtschaftssystem zugrunde liegt. Der Aufruf des PHP-Skriptes erfolgt in dem Shell-Skript *build_atAusgang.sh*. Dieses Skript wird über das Shell-Skript *atAusgang.sh* aufgerufen, welches über einen *cronjob* gestartet wird. Des Weiteren beinhaltet das Shell-Skript *atAusgang.sh* den Aufruf des Skriptes *copy_atAusgang.sh*, welches für den Abgleich der Änderungsdaten dieser Funktion zuständig ist.

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3 echo "START: `date`"
4 echo "##### aufruf_build_atAusgang.sh #####"
5 sh $shellpfad_atAusgang/build_atAusgang.sh
6 echo "##### aufruf_copy_atAusgang.sh #####"
7 sh $shellpfad_atAusgang/copy_atAusgang.sh
8 echo "ENDE: `date`"
```

Quelltext 4.2: Aufruf zweier Shell-Skripte in atAusgang.sh

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3 echo "##### Build_atAusgang_auf_NEXUS-NEU #####"
4 echo "-----STEP_1_SELECT_BELEGE-----"
5 $phppfad/php $phppfad_atAusgang/select_beleg_ausgang.php > $logpfad_php/
   select_beleg_ausgang.log
6 echo "-----STEP_2_CLEAR_HNR-----"
```

Quelltext 4.3: Aufruf eines PHP-Skriptes in build_atAusgang.sh

Das PHP-Skript (siehe Quelltext im Anhang D.1 auf CD-ROM Seite 97) zur Ermittlung der Daten aus den Angeboten im Warenwirtschaftssystem muss neu geschrieben werden. Dabei sind folgende Bedingungen zu erfüllen:

- Ermittlung aller Belege, welche mit *@Ausgang* gekennzeichnet sind
- Ermittlung aller Positionen aus den markierten Belegen
- Ermittlung der Herstellernummer sowie der Netto- bzw. Bruttopreise aller Positionen
- Feststellen, ob jede interne Artikelnummer eine Herstellernummer besitzt
- Einfügen der ermittelten Positionsdaten in die Hilfstabelle *@Ausgang* im Intranet
- Löschen von Belegen aus der Hilfstabelle, welche nicht mehr mit *@Ausgang* gekennzeichnet sind
- Löschen von Belegen aus der Hilfstabelle, welche nicht mehr im Zeitschlauch sind
- Löschen von Positionen aus der Hilfstabelle, welche nicht mehr im Angebot vorhanden sind
- Versenden einer E-Mail mit Daten über Positionen, welche keine Herstellernummer besitzen

Nach der Implementierung, der oben genannten Bedingungen, muss die Hilfstabelle *@Ausgang* noch einen weiteren Algorithmus durchlaufen. Dazu dient die Datei *clear_hnr.php* (siehe Quelltext im Anhang D.2 auf CD-ROM Seite 104). Mit dem PHP-Skript *clear_hnr.php* werden einem Kunden doppelt zugewiesene Herstellernummern gelöscht. Dies kann auftreten, wenn einem Kunden mehrere Angebote zugeordnet sind und diese einen Artikel beinhalten, der auch in einem anderen Angebot existiert. Es haben nur der Preis und die Herstellernummer Bestand, welche aus dem aktuellsten Angebot stammen.

Die so gewonnenen und nach alle Kriterien bereinigten Daten werden in der Hilfstabelle *@Ausgang* im Intranet gesammelt und gespeichert. Diese Daten müssen mit den bestehenden Daten aus dem Web-Shop abgeglichen und evtl. aktualisiert werden.

4.3.2 Ermittlung der Änderungsdaten

Die Ermittlung der Änderungsdaten wird durch den Aufruf des Shell-Skriptes *copy_atAusgang.sh* (siehe Quelltext im Anhang D.4 auf CD-ROM Seite 106) in der *atAusgang.sh* initiali-

siert. Das Skript *copy_atAusgang.sh* realisiert dabei folgenden Ablauf:

- Ermittlung der Daten aus @Ausgang im Web-Shop-System
- Erstellung einer CSV-Datei anhand der Daten aus dem Web-Shop-System (Slave)
- Ermittlung der Daten aus dem Intranet
- Erstellung einer CSV-Datei anhand der Daten aus dem Intranet (Master)
- Erstellung der Differenz-Datei
- Erstellung einer ADD- bzw. REM-Datei aus der Differenz-Datei
- kopieren der ADD- bzw. REM-Datei über *scp* in das Web-Shop-System
- löschen der erstellten und nicht mehr benötigten Dateien

4.3.3 Übertragung der Änderungsdaten in das Web-Shop-System

Die Übertragung der ADD- bzw. REM-Datei wird ebenfalls in dem Shell-Skript *copy_atAusgang.sh* realisiert. Die Übertragung wird mit den Befehlen

```
scp $homepfad_M2S/atAusgang_diff_add.txt prosoft@trinity:/home/prosoft/tmp/  
atAusgang_diff_add.txt
```

```
scp $homepfad_M2S/atAusgang_diff_rem.txt prosoft@trinity:/home/prosoft/tmp/  
atAusgang_diff_rem.txt
```

umgesetzt. Mit *scp* werden die ADD- bzw. REM-Dateien über eine gesicherte Verbindung in das Web-Shop-System kopiert. Somit stehen sie zu weiteren Verarbeitung im Web-Shop-System zu Verfügung.

Nachdem die ADD- bzw. REM-Dateien in das Web-Shop-System übertragen sind, wird in dem Skript *copy_atAusgang.sh* der *SSH*-Befehl

```
ssh prosoft@trinity "php /home/prosoft/bin/atAusgang_update.php"
```

ausgeführt.

Dieser Befehl ruft im Web-Shop-System ein PHP-Skript *atAusgang_update.php* auf, welches den Import der ermittelten und übertragenen Daten durchführt.

4.3.4 Import der Änderungsdaten in die Web-Shop-Datenbank

Der Import der übertragenen Änderungsdaten wird durch das PHP-Skript *atAusgang_update.php* (siehe Quelltext im Anhang D.5 auf CD-ROM Seite 107) im Web-Shop-System realisiert. In diesem PHP-Skript werden die vorher übertragenen ADD- bzw. REM-Dateien eingelesen und daraus die zu ändernden, zu löschenden und hinzuzufügenden Daten ermittelt. Mit geeigneten SQL-Befehlen wird die *@Ausgang*-Tabelle im Web-Shop manipuliert.

Nachdem alle Daten in der Web-Shop-Datenbank manipuliert sind, wird noch das Tabellenfeld *last_update* mit dem aktuellen Datum und Zeit gesetzt. Dies ist notwendig, um bei einer laufenden Benutzersitzung im Web-Shop das Session-Objekt zu löschen und mit den aktuellen *@Ausgang*-Daten zu füllen.

Nach der Abarbeitung des PHP-Skriptes *atAusgang_update.php* ist auch das Shell-Skript *copy_atAusgang.sh* und somit das Shell-Skript *atAusgang.sh* durchlaufen. Nun stehen die aktuellen Daten aus dem Warenwirtschaftssystem dem Web-Shop zur Anzeige zur Verfügung. Nach einer erfolgreichen Verarbeitung des Shell-Skriptes *atAusgang.sh* sieht die Log-Datei wie folgt aus.

```
1 START: Fri Jan 30 15:50:00 CET 2009
2 ##### aufruf build_atAusgang.sh #####
3 ##### Build atAusgang auf NEXUS-NEU #####
4 -----STEP 1 SELECT_BELEGE-----
5 -----STEP 2 CLEAR_HNR-----
6 ##### aufruf copy_atAusgang.sh #####
7 ENDE: Fri Jan 30 15:50:02 CET 2009
```

Quelltext 4.4: Log-Datei nach abarbeiten der *atAusgang.sh*

4.3.5 Anzeige der Daten im Web-Shop

Die übertragenen Daten aus dem Warenwirtschaftssystem müssen dem Benutzer zugänglich gemacht werden. Dazu müssen diese aus der Web-Shop-Datenbank ermittelt und im Web-Shop dargestellt werden. Dieser Abschnitt befasst sich mit der Umsetzung dieser Problematik.

Wie im Abschnitt 3.4.2 auf Seite 42 erläutert, werden alle Daten aus der Web-Shop-Datenbank in ein Session-Objekt gespeichert. Somit müssen diese nur einmal abgefragt werden und stehen während einer Sitzung des Benutzers im Web-Shop zur Verfügung.

Bevor einem Benutzer des Web-Shops diese Funktionalität zur Verfügung steht, muss sich dieser in den Web-Shop über sein vergebenes Login und Passwort einloggen. Ist der Benutzer dem System unter den eingegebenen Daten bekannt, wird überprüft, ob dieser Kunde die Funktionalität für kundenspezifische Preise nutzen darf. Ist dies der Fall, wird ein Flag zur späteren Auswertung gesetzt. Diese Überprüfung findet in der *makeID.php* statt und wird anhand des folgenden Quelltextauszugs ersichtlich.

```

1 .
2 .
3 .
4 // Abfrage der User-Daten
5 $res=mysql_query("SELECT UserID, Name, Vorname, Preistyp, Nummer, AlleArtikel,
    AnzPS, MentionKundennummer, MentionBez, atAusgang, Favoritenliste,
    OnlineBestellung, Anfrage, KeinBestellen FROM $kundentable WHERE (Login='
    $Login') AND (Passwort=MD5('$Passwort'))", $dbh1);
6 .
7 .
8 .
9 //ob der eingeloggte User atAusgang-Preise von Belegen aus Merntion sieht
10 $_SESSION['hat_atAusgang']=mysql_result($res,0,9);
11 .
12 .
13 .

```

Quelltext 4.5: Überprüfung der Logindaten und der *@Ausgang*-Funktionalität

Die Abfrage der Daten für die kundenspezifischen Preise wird nach erfolgreichem Login in der *index.php* (siehe Quelltext im Anhang D.6 auf CD-ROM Seite 108) des Web-Shops durchgeführt. Dafür dient folgender Quelltextauszug.

```

1 .
2 .
3 //Nur einmal bis Session-Objekt atAusgang angelegt ist (wird beim abmelden in
    abmelden.php geleert)
4 if(!isset($_SESSION['atAusgang'])){
5     //Abfrage der atAusgangsTabelle nach HNR und Preisen für die eingeloggte
        MentionKundennummer aus KundenTabelle anhand des Logins
6     $query_atAusgang = "SELECT_hnr, bpvkpreis, bpvkbrutt FROM
        $atausgangtable WHERE bskundennr='$_SESSION[MentionKdnr]'";
7
8     $result_atAusgang = mysql_query($query_atAusgang, $dbh1);
9

```

```

10      //wenn Abfrage mindestens ein Ergebnis liefert Objekt erstellen
11      if(mysql_num_rows($result_atAusgang)!=0){
12          //erstellt ein SESSION-Objekt mit allen Herstellernummern,
              Preisen, Bruttopreisen die der Eingeloggten $_SESSION[
              MentionKdnr] zugeordnet sind
13          //mit dem jungsten letzten Aenderungsdatum
14          $_SESSION['atAusgang'] = new atAusgang(mysql_result(mysql_query(
              "SELECT_max(last_update)_FROM_ $atausgangtable" , $dbh1), 0, 0));
15
16          .
17          .
18      }
19  }
20  .
21  .

```

Quelltext 4.6: Auszug der *index.php* für die Funktion *@Ausgang*

Es stehen nun alle notwendigen Daten für die Anzeige von kundenspezifischen Preisen innerhalb des Erneuerungszeitraumes in dem Session-Objekt zur Verfügung und müssen nicht neu ermittelt werden.

Alle Daten für die kundenspezifischen Preise eines eingeloggten Benutzers werden in dem Session-Objekt als Array gespeichert. Die Daten können mit Objektmethoden gelesen, geschrieben und verändert werden. Das Klassendiagramm im Anhang C.2 auf Seite 95 bietet einen Überblick über die verwendeten Methoden.

Die benötigten Daten für die Darstellung der kundenspezifischen Preise sind die Herstellernummer des Artikels, der Netto- und Bruttopreis des Angebotes, der Netto- und Bruttopreis im Web-Shop sowie die Zeit der letzten Datenaktualisierung.

Validierung der Angebotspreise

Die Anzeige von kundenspezifischen Preisen aus einem Angebot ist für eine günstigere Preisgestaltung gedacht. Der Benutzer soll dadurch einen Artikel günstiger erhalten, als dies für einen Benutzer ohne ein spezielles Angebot der Fall ist. Es kann aber im Verlauf eines bestehenden, mit @Ausgang gekennzeichneten, Angebots dazu kommen, dass sich der Produktpreis dahin gehend entwickelt, dass dieser günstiger ist als der Preis im Angebot. Ist dies der Fall, soll der günstigere Listenpreis Vorrang haben, da sonst der Kunde mit einem älteren und teureren Preis benachteiligt wird. Somit ist es im Vorfeld notwendig, die Angebotspreise im Vergleich zu den aktuellen Listenpreisen zu prüfen.

Der Vergleich wird nach dem Füllen des Arrays im Session-Objekt in der *index.php* durchgeführt. Dazu wird die implementierte Funktion *checkpreis()* aufgerufen. Der Funktion wird der Preistyp des eingeloggten Benutzers, die Herstellernummer und der Index des Arrays übergeben. Die Funktion zur Prüfung des aktuellen Artikelpreises wird durch den Befehl

```
//aufruf der Funktion zum vergleich des atAusgangspreises mit dem Shoppreis
$_SESSION['atAusgang']->checkpreis($_SESSION['Preis'],$row[0],$i);
```

aufgerufen. Der Quelltextauszug 4.7 verdeutlicht die Funktionsweise.

```
1 //ermittelt ob der atAusgangspreis größer ist als der ShopPreis
2 function checkpreis($preistyp,$hnr,$element){
3     //wenn Preis (also Endkunde mit Bruttopreis)
4     if($preistyp=="Preis"){
5         //wenn Shoppreis Brutto < atAusgangspreis Brutto dann Shoppreis in
           atAusgang-arr speichern
6         if($this->shop_preis_arr[$hnr]<$this->brutto_arr[$element]){
7             //weist dem atAusgangsObjekt den kleineren ShopPreis zu
8             $this->brutto_arr[$element]=$this->shop_preis_arr[$hnr];
9         }
10    }
11    else{ //wenn Preis2 (also Händler mit Nettopreis)
12        //wenn Shoppreis Netto < atAusgangspreis Netto dann Shoppreis in
           atAusgang-arr speichern
13        if($this->shop_preis2_arr[$hnr]<$this->preis_arr[$element]){
14            //weist dem atAusgangsObjekt den kleineren ShopPreis zu
15            $this->preis_arr[$element]=$this->shop_preis2_arr[$hnr];
16        }
17    }
18 }
```

Quelltext 4.7: Funktion zum Vergleich des Angebotspreises mit dem Web-Shoppreis

Anzeige der Angebotspreise

Die Anzeige der Angebotspreise findet in den Artikellisten sowie in der Detailansicht des Web-Shops statt. Dafür wird an der Stelle im Skript, an der der Artikelpreis anhand der Herstellernummer ermittelt wird, überprüft, ob es sich um einen Angebotsartikel handelt. Ist dies der Fall, wird der Preis aus dem Web-Shop mit dem Preis aus dem

Session-Objekt überschrieben. Der nachstehende Quelltextauszug zeigt die Vorgehensweise.

```
1 //wenn ein Session-Objekt erstellt ist
2 if(isset($_SESSION['atAusgang'])) {
3     //schaut ob HNR im Objekt vorhanden ist
4     if($_SESSION['atAusgang']->is_in_array($hnr)) {
5         //ermittelt Array Key von der HNR
6         $key=$_SESSION['atAusgang']->get_hnr_arr_key($hnr);
7
8         //setzten eines Flags für spätere Entscheidung (ist EIN atAusgang
9         Artikel)
10        $atAusgang="TRUE";
11
12        //Benutzer sieht Netto-Preise
13        if($_SESSION['Preis']=="Preis2") {
14            //Netto-Preis aus Objekt
15            $Preis=$_SESSION['atAusgang']->get_preis($key);
16        }
17        else { //Benutzer sieht Brutto-Preise
18            //Brutto-Preis aus Objekt
19            $Preis=$_SESSION['atAusgang']->get_brutto($key);
20        }
21    }
22    else { //HNR NICHT Objekt vorhanden
23        //setzten eines Flags für spätere Entscheidung (ist KEIN atAusgang
24        Artikel)
25        $atAusgang="FALSE";
26    }
27 }
```

Quelltext 4.8: Ermittlung des Angebotspreises für die Anzeige in der Artikelliste

Ist der betreffende Artikel ein Angebotsartikel, enthält die Variable *\$Preis* den Angebotspreis, ansonsten den Web-Shop-Preis. Anhand des Flags *\$atAusgang* wird dieser Artikel für eine spätere Verarbeitung als Angebotsartikel gekennzeichnet. Diese Verarbeitung findet in dem nachstehenden Quelltextauszug statt.

```
if($atAusgang=="TRUE"){
    $table.="<a title=\"Ihr Angebotsartikel\">
        <img src=\"images/ang_gruen_inverse_klein.bmp\"></a>";
}
```

Hierbei wird der Angebotsartikel optisch als dieser in der Artikelliste gekennzeichnet. Der Vergleich zwischen einem Angebotsartikel und einem Web-Shop-Artikel wird auf den Abbildungen 4.3 bzw. 4.4 auf Seite 56 dargestellt. Darin wird die optische Kennzeichnung des Angebotsartikels (siehe Abbildungen roter Pfeil) sowie der Preisunterschied (siehe Abbildungen grüner Pfeil) ersichtlich.

Suchergebnis zu: 'APPLE Mighty Mouse Wireless - Maus - Laser'

Treffer in den Kategorien

☐ Peripherie (1)

Erweiterte Suche

Hersteller: Alle
Suche: Info APPLE Mighty Mouse Wireless - Maus - Laser
Suchfelder: Info ArtNr. Bezeichnung HerstNr.
Artikel pro Seite: 100 (min. 5 max. 100)
Suche starten Filter zurücksetzen

Nur ProSoft-Artikel: Info
Details anzeigen: Info
Nur Promotions: Info

Ihre Suchanfrage ergab 1 Treffer

* Alle Artikel-Preise sind inklusive 19% MwSt. und zuzüglich Versandkosten.

ArtNr.	Bezeichnung ▲	Preis*	HerstellerNr.
58025	APPLE Mighty Mouse Wireless - Maus - Laser	82.00 €*	MB1112M/A

erste | vorherige | nächste | letzte nach oben

* Alle Artikel-Preise sind inklusive 19% MwSt. und zuzüglich Versandkosten.

Abbildung 4.3: Web-Shopartikel ohne Angebotspreis

Suchergebnis zu: 'APPLE Mighty Mouse Wireless - Maus - Laser'

Treffer in den Kategorien

☐ Peripherie (1)

Erweiterte Suche

Hersteller: Alle
Suche: Info APPLE Mighty Mouse Wireless - Maus - Laser
Suchfelder: Info ArtNr. Bezeichnung HerstNr.
Artikel pro Seite: 100 (min. 5 max. 100)
Suche starten Filter zurücksetzen

Nur ProSoft-Artikel: Info
Details anzeigen: Info
Nur Promotions: Info

Ihre Suchanfrage ergab 1 Treffer

* Alle Artikel-Preise sind inklusive 19% MwSt. und zuzüglich Versandkosten.

ArtNr.	Bezeichnung ▲	Preis*	HerstellerNr.
58025	APPLE Mighty Mouse Wireless - Maus - Laser	66.00 €*	MB1112M/A

erste | vorherige | nächste | letzte nach oben

* Alle Artikel-Preise sind inklusive 19% MwSt. und zuzüglich Versandkosten.

Abbildung 4.4: Web-Shopartikel mit Angebotspreis

Die Funktionalität der kundenspezifischen Preise kann theoretisch für beliebig viele Benutzer des Web-Shops, mit beliebig vielen Angeboten und unbegrenzter Anzahl der darin enthaltenen Positionen genutzt werden. Dies wird sich jedoch in der Praxis auf ausgewählte Kunden der ProSoft Krippner GmbH beschränken.

Aktualisierung der Daten während einer Benutzersitzung

Wie im Abschnitt 3.4.1 auf Seite 41 beschrieben, müssen die Daten während einer Sitzung des Benutzers im Web-Shop neu aus der Web-Shop-Datenbank gelesen werden. Dazu dient das Datenbankfeld *last_update*. Dieses Feld wird am Ende des Datenimports durch *atAusgang_update.php* gesetzt, und enthält somit den letzten Aktualisierungszeitpunkt der Daten. In der *index.php* wird bei einem erstellten Session-Objekt überprüft, ob das Delta zwischen aktueller Zeit und dem Aktualisierungszeitpunkt größer ist als das Intervall der Aktualisierung.

```
1 .
2 .
3 //wenn sich ein User laenger als 10 Minuten im Shop bewegt
4 //wird das Session-Objekt gelöscht um es mit den aktuellsten Daten wieder zu
  füllen
5 if(isset($_SESSION['atAusgang'])) {
6
7     //ueberprueft die Aktualitaet des Objektes (aufruf eine Funktion)
8     if(CheckAktualitaet(($time_to_akt_at_ausgang*60),$_SESSION['atAusgang']->
        get_max_last_mod())) {
9         //löscht das Session-Objekt
10        unset($_SESSION['atAusgang']);
11    }
12 }
13 .
14 .
```

Quelltext 4.9: Initialisierung der Datenaktualisierung für @Ausgang

Ist dies der Fall, wird das Session-Objekt zerstört und mit den aktuellen Daten aus der Datenbank neu erstellt. Dies realisiert die implementierte Funktion *CheckAktualitaet()* (siehe Quelltext im Anhang D.7 auf CD-ROM Seite 114)

4.4 Favoritenliste

Wie im Verlauf dieser Arbeit erwähnt, können die Funktionen der kundenspezifischen Preise und der Favoritenliste analog betrachtet werden. Sie besitzen die gleichen Anforderungen an die Abfragen im Warenwirtschaftssystem, der Datenhaltung in den Hilfstabellen und der Datenübertragung in den Web-Shop. Somit wird bei der Realisierung nur kurz auf diese Punkte eingegangen. Die Darstellung der Daten im Web-Shop wird

ausführlicher beschrieben, da sich diese von den kundenspezifischen Preisen unterscheidet.

4.4.1 Ermittlung der Daten aus dem Warenwirtschaftssystem

Die Ermittlung der Daten aus dem Warenwirtschaftssystem erfolgt analog den Angebotspreisen. Dazu wird über die Jobsteuerung das Shell-Skript *listeFavoriten.sh* (siehe Quelltext im Anhang D.8 auf CD-ROM Seite 114) ausgeführt. Dieses Skript führt nacheinander die Shell-Skripte *build_MentionFavoriten.sh* (siehe Quelltext im Anhang D.9 auf CD-ROM Seite 115) und *copy_MentionFavoriten.sh* (siehe Quelltext im Anhang D.10 auf CD-ROM Seite 115) aus.

Das in dem Skript *build_MentionFavoriten.sh* aufgerufene PHP-Skript *select_liste_favoriten.php* (siehe Quelltext im Anhang D.11 auf CD-ROM Seite 116) ermittelt alle notwendigen Daten aus dem Warenwirtschaftssystem und unterliegt, wie bei den kundenspezifischen Preisen, gewissen Anforderungen. Folgende Aufgaben muss das Skript erfüllen:

- Ermittlung aller Belege, die mit *langfristiges Angebot* gekennzeichnet sind
- Ermittlung aller Positionen aus den markierten Belegen
- Ermittlung der Herstellernummer aller Positionen
- Feststellen, ob jede interne Artikelnummer eine Herstellernummer besitzt
- Einfügen der ermittelten Positionsdaten in die Hilfstabelle *MentionFavoriten* im Intranet
- Löschen von Belegen aus der Hilfstabelle, welche nicht mehr mit *langfristiges Angebot* gekennzeichnet sind
- Löschen von Belegen aus der Hilfstabelle, welche nicht mehr im Zeitschlauch sind
- Löschen von Positionen aus der Hilfstabelle, welche nicht mehr im Angebot vorhanden sind
- Versenden einer E-Mail mit Daten über Positionen, die keine Herstellernummer besitzen

Die erstellte Hilfstabelle *MentionFavoriten* darf wie bei den kundenspezifischen Preisen nur jeweils eine Herstellernummer eines Artikels für einen Kunden besitzen. Folglich muss auch diese Tabelle dahin gehend bereinigt werden. Dies geschieht über das PHP-

Skript *clear_hnr.php* (siehe Quelltext im Anhang D.12 auf CD-ROM Seite 123), welches nach Abarbeitung des PHP-Skriptes *select_liste_favoriten.php* inkludiert wird.

4.4.2 Ermittlung der Änderungsdaten

Wie bei den kundenspezifischen Preisen erfolgt eine Änderungsdatenermittlung (siehe Quelltext im Anhang D.10 auf CD-ROM Seite 115) über die UNIX-Funktionalität *diff*. Dabei wird bei der Favoritenliste vergleichbar zu den Angebotspreisen vorgegangen:

- Ermittlung der Daten aus MentionFavoriten im Web-Shop-System
- Erstellung einer CSV-Datei anhand der Daten aus dem Web-Shop-System (Slave)
- Ermittlung der Daten aus dem Intranet
- Erstellung einer CSV-Datei anhand der Daten aus dem Intranet (Master)
- Erstellung der Differenz-Datei
- Erstellung einer ADD- bzw. REM-Datei aus der Differenz-Datei
- Kopieren der ADD- bzw. REM-Datei über *scp* in das Web-Shop-System
- Löschen der erstellten und nicht mehr benötigten Dateien

4.4.3 Übertragung der Änderungsdaten in das Web-Shop-System

In dem Shell-Skript *copy_MentionFavoriten.sh* wird die Übertragung der Änderungsdaten realisiert. Dazu werden die Befehle

```
scp $homepfad_M2S/MentionFavoriten_diff_add.txt prosoft@trinity:/home  
/prosoft/tmp/MentionFavoriten_diff_add.txt
```

```
scp $homepfad_M2S/MentionFavoriten_diff_rem.txt prosoft@trinity:/home  
/prosoft/tmp/MentionFavoriten_diff_rem.txt
```

ausgeführt.

4.4.4 Import der Änderungsdaten in die Web-Shop-Datenbank

Ebenfalls zu vergleichen ist der Import der Daten für die Favoritenliste in die Web-Shop-Datenbank. Um diesen Import durchzuführen, wird das PHP-Skript *MentionFavoriten_update.php* (siehe Quelltext im Anhang D.13 auf CD-ROM Seite 124) im Shell-Skript *copy_MentionFavoriten.sh* aufgerufen und ausgeführt. Das PHP-Skript *MentionFavoriten_update.php* liest die vorher übertragenen ADD- bzw. REM-Dateien ein und manipuliert mit geeigneten SQL-Befehlen die MentionFavoriten-Tabelle im Web-Shop. Zuletzt wird das Tabellenfeld *last_update* mit dem aktuellen Datum und der aktuellen Zeit gesetzt.

4.4.5 Anzeige der Daten im Web-Shop

Die zur Anzeige der Favoritenliste aus dem Warenwirtschaftssystem notwendigen Daten werden analog zu den Daten für die kundenspezifischen Preise, nach einem erfolgreichen Login eines dafür freigeschalteten Benutzers, in einem Session-Objekt gespeichert. Wie bei der Ermittlung der Angebotspreisedaten muss der Benutzer des Web-Shops für diese Funktionalität freigeschaltet sein. Die Freischaltung wird im Vorfeld in der *makeID.php* ermittelt (siehe Quelltextauszug 4.10 auf Seite 60). Ist der Benutzer für die Nutzung einer Favoritenliste eingetragen, wird die Variable `$_SESSION['hat_Favoritenliste']` gesetzt.

```
1 .
2 .
3 // Abfrage der User-Daten
4 $res=mysql_query("SELECT UserID, Name, Vorname, Preistyp, Nummer, AlleArtikel,
    AnzPS, MentionKundennummer, MentionBez, atAusgang, Favoritenliste,
    OnlineBestellung, Anfrage, KeinBestellen FROM $kundentable WHERE (Login='
    $Login') AND (Passwort=MD5('$Passwort'))", $dbh1);
5 .
6 .
7 .
8 //ob der eingeloggte User eine aus Mention gesteuerte Favoritenliste sieht
9 $_SESSION['hat_Favoritenliste']=mysql_result($res,0,10);
10 .
11 .
```

Quelltext 4.10: Überprüfung der Logindaten und der Favoritenlisten-Funktionalität

Die Ermittlung der Daten findet in der *index.php* statt und wird durch folgenden Quelltextauszug realisiert.

```

1 //Nur einmal bis Session-Objekt listeFavoriten angelegt ist
2 //(wird beim abmelden in abmelden.php geleert und wenn ein Artikel hinzugefuegt
   wurde bzw entfernt wurde)
3 if(!isset($_SESSION['listeFavoriten'])){
4
5     //wenn User für Favoritenliste aus Mention freigeschaltet ist
6     if($_SESSION['hat_Favoritenliste']==1){
7         //ermittelt alle HNR aus MentionFavoriten && ShopFavortien des Users
8         $query_Favoritenliste = "(SELECT_hnr_FROM_ $shopfavoritentable_WHERE_
           UserID='$_SESSION[UserID]')_UNION_(SELECT_hnr_FROM_
           $mentionfavoritentable_WHERE_ bskundennr='$_SESSION[MentionKdnr]')";
9     }
10    else{
11        //ermittelt alle HNR ShopFavortien des Users
12        $query_Favoritenliste = "SELECT_hnr_FROM_ $shopfavoritentable_WHERE_
           UserID='$_SESSION[UserID]'";
13    }
14
15    $result_Favoritenliste = mysql_query($query_Favoritenliste);
16
17    //wenn der angemeldete User mindesten 1 Artikel in Favoritenliste (Shop bzw.
       Mention) besitzt
18    if(mysql_num_rows($result_Favoritenliste)>0){
19        //erstellt ein SESSION-Objekt listeFavoriten mit allen Herstellernummern
20        //die der Eingeloggten User als Favoriten in Shop bzw. MentionFavoriten
           eingetragen hat
21        //mit dem jungsten letzten Aenderungsdatum
22        $_SESSION['listeFavoriten'] = new listeFavoriten(mysql_result(
           mysql_query("SELECT_max(last_update)_FROM_ $mentionfavoritentable",
           $dbh1),0,0));
23
24        //Füllt das Session-Objekt listeFavoriten mit Werten aus dem SELECT
25        while($row = mysql_fetch_array($result_Favoritenliste)){
26            $_SESSION['listeFavoriten']->add_hnr($row['hnr']);
27        }
28    }
29 }

```

Quelltext 4.11: Auszug der *index.php* für die Funktion *Favoritenliste*

Das Klassendiagramm im Anhang C.1 auf Seite 94 zeigt die bei der Umsetzung verwendeten Methoden, um die notwendigen Daten in dem Session-Objekt zu speichern, zu lesen oder zu verändern.

Im Gegensatz zu den kundenspezifischen Preisen, benötigt die Funktion der Favoritenliste lediglich die Herstellernummer des Artikels sowie die Zeit der letzten Datenaktualisierung.

Darstellung der Favoritenliste

Um auf die Favoritenliste zugreifen zu können, bedarf es eines neuen Menüpunktes. Der Menüpunkt "Mein ProSoft" wird in der Template-Datei *index.html* erstellt.

```

1 <div id="main_menu_menu">
2     <table style="float:right;">
3         <tr>
4             {mein_prosoft}
5             <td class="main_menu_no_padding_left"><a href="{https}/index.php?
              tpl_var={an_ab_seite}">{An_Ab}</a></td>
6             <td class="main_menu_trennung"></td>
7             <td class="main_menu" style="padding-left:24px;padding-right:10px;"
              ><a href="{https}/index.php?tpl_var={zweites_menu}&wgleeren
              =1">{Reg_Konto}</a></td>
8             <td class="main_menu_trennung"></td>
9             <td class="main_menu"><a href="{http}/index.php?tpl_var=hilfe&
              wgleeren=1&sucheleeren=3">Hilfe </a></td>
10            <td class="main_menu_trennung"></td>
11            <td class="main_menu"><a href="{http}/index.php?tpl_var=agb&
              wgleeren=1&sucheleeren=3">AGB</a></td>
12            <td class="main_menu_trennung"></td>
13            <td class="main_menu"><a href="{http}/index.php?tpl_var=kontakt&
              wgleeren=1&sucheleeren=3">Kontakt </a></td>
14            {service}
15        </tr>
16    </table>
17 </div>

```

Quelltext 4.12: Menücontainer in der Template-Datei *index.php* für den Menüpunkt "Mein ProSoft"

An der Stelle des Platzhalters *mein_prosoft* wird nach erfolgreichem Login des Benutzers der Menüpunkt "Mein ProSoft" erstellt. Das Ersetzen des Platzhalters, auch

parsen genannt, wird in der *template.php* realisiert. Der nachstehende Quelltextauszug soll dies zeigen.

```
1 //überprüfen, ob Template-Variable begreussung_kopf gesetzt
2 if (isset($template_mein_prosoft)){
3     $tpl->set_file("mein_prosoft", $template_mein_prosoft);
4
5     //wenn User angemeldet ist
6     if($UserKunde==1){
7         $tpl->parse("temp", "mein_prosoft");
8     }
9 }
```

Quelltext 4.13: Ersetzen des Platzhalters für den Menüpunkt "Mein ProSoft"

Damit der Platzhalter mit dem HTML-Inhalt ersetzt werden kann, muss dieser Inhalt in einer Template-Datei erstellt werden. Dafür dient die Datei *mein_prosoft.html*.

```
1 <td class="main_menu_no_padding_left">
2     <a href="{http}/index.php?tpl_var=mein_prosoft&sucheleeren=4&
3         carepackleeren=2&kommevonleeren=1">Mein ProSoft </a>
4 </td>
5 <td class="main_menu_trennung" style="padding-right:24px;"></td>
```

Quelltext 4.14: Inhalt der Template-Datei *mein_prosoft.html* für den Menüpunkt "Mein ProSoft"

Diese Template-Datei muss vorher in der *index.php* durch den Befehl

```
//Titel der Seite die Anstelle der Teplatevariable {mein_prosoft}
in der index.html eingefügt wird
$template_mein_prosoft="sites/mein_prosoft.html";
```

einer Variablen übergeben werden.

Die nachstehenden Abbildungen zeigen den Menüpunkt "Mein ProSoft" (siehe Abbildung 4.5 auf Seite 64 bzw. Abbildung 4.6 auf Seite 64 roter Pfeil) in dem Web-Shop-Menü vor bzw. nach einem erfolgreichen Login.



Abbildung 4.5: Web-Shop-Menü ohne "Mein ProSoft"



Abbildung 4.6: Web-Shop-Menü mit "Mein ProSoft"

Über den erstellten Menüpunkt "Mein ProSoft" kann der Benutzer auf die Favoritenliste zugreifen. Zuvor wird überprüft, ob Artikel für die Darstellung einer Favoritenliste zur Verfügung stehen. Die Überprüfung findet in dem zu erstellenden PHP-Skript *mein_prosoft_auswahl.php* (siehe Quelltext im Anhang D.14 auf CD-ROM Seite 126) statt. Stehen dem Benutzer noch keine Artikel für die Favoritenliste zur Verfügung, wird dies angezeigt (siehe Abbildung 4.7 roter Pfeil).

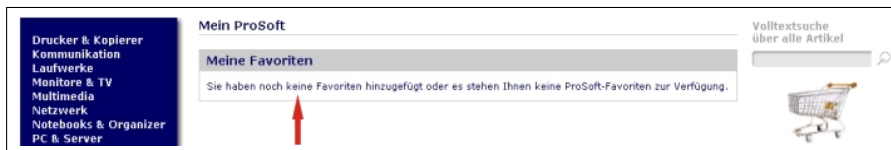


Abbildung 4.7: "Mein ProSoft" ohne Auswahl einer Favoritenliste

Sind Artikel für eine Favoritenliste vorhanden, so kann der Benutzer diese einsehen (siehe Abbildung 4.8 grüner Pfeil).



Abbildung 4.8: "Mein ProSoft" mit Auswahl einer Favoritenliste

Die für den Benutzer verfügbaren Favoriten werden in einer Artikelliste (siehe Abbildung 4.9 auf Seite 65) dargestellt.



Abbildung 4.9: Favoritenliste mit Darstellung von Artikeln

Die Funktionalität der Favoritenlisten wird in dem PHP-Skript *mein_prosoft_favoriten.php* implementiert. Da das Design und der Funktionsumfang der Auflistung von Artikeln gleicht, konnte auf die Implementierung der Artikelliste zurückgegriffen werden.

Da alle Favoriten des Benutzers reale Artikel im Web-Shop sind, muss eine Verbindung zwischen den Herstellernummern im Session-Objekt und der Artikeltabelle in der Web-Shop-Datenbank hergestellt werden. Dies realisiert der folgende Quelltextauszug aus dem PHP-Skript *mein_prosoft_favoriten.php*.

```

1 .
2 .
3 .
4 //leert vor dem bilden der Temporären Tabelle die alte TempFavoriten-Tabelle
5 $abfrage_truncate="TRUNCATE_TABLE_ $tempfavoritentable";
6
7 mysql_query($abfrage_truncate,$dbh1);
8
9 $select_favoriten_shop="INSERT INTO_ $tempfavoritentable_ (SELECT_O. Bezeichnung_ as
   _Bezeichnung, W.Name_ as_ WarengruppenName, A.hnr, H.Name_ AS_ HerstellerName, H.
   Nummer_ AS_ HerstellerID, W.AnzWeb_ as_ WarengruppeAnz, W.Nummer, W.OberWG_FROM_
   $wgtable_ AS_ W, _ $artikeltabelle_ AS_ A, _ $oberwgtable_ as_ O, _ $hersttable_ AS_ H WHERE
   _";
10
11 $i=0;
12
13 if(count($_SESSION['listeFavoriten']->hnr_arr)>0){
14     foreach($_SESSION['listeFavoriten']->hnr_arr as $key => $hnr_favorit){
15         if($i==0){

```

```

16         $select_favoriten_shop="(A.hnr='$hnr_favorit '";
17     }
18     else{
19         $select_favoriten_shop="_OR_A.hnr='$hnr_favorit '";
20     }
21     $i++;
22 }
23 }
24 $select_favoriten_shop="_OR_A.hnr='' )_AND_A.Warengruppe=W.Nummer_AND_A.
    Hersteller=H.Nummer_AND_O.Nummer=W.OberWG_AND_A.ausbl!='TRUE' _AND_A.eol!='
    E'_$bbw_$anzeige_ps_artikel_$anzeige_promo_artikel_$suchquery_$nach_oberwg_
    $nach_wg_$nach_Hersteller_$kundenartikel_ORDER_BY_Bezeichnung ,
    WarengruppenName)";
25 .
26 .
27 .

```

Quelltext 4.15: Erstellen einer temporären Tabelle für die Favoritenliste

Anhand der Herstellernummer aus dem Session-Objekt wird eine temporäre Datenbanktabelle *TempFavoriten* (siehe Anhang B.3 auf Seite 93) erstellt, die alle benötigten Daten aus der "Mastertabelle" (siehe Anhang B.2 auf Seite 92) enthält. Somit stehen alle Daten, die für die Bildung und den Funktionsumfang der Favoritenliste notwendig sind, direkt zur Verfügung. Dies wirkt sich stark auf die Performance aus, da somit keine Selects über mehrere Tabellen notwendig sind.

Aktualisierung der Daten während einer Benutzersitzung

Die Aktualität der Daten wird wie bei den kundenspezifischen Preisen (siehe Abschnitt 4.3.5 auf Seite 57) mit Hilfe der Funktion *CheckAktualitaet()* (siehe Quelltext im Anhang D.7 auf CD-ROM Seite 114) überprüft.

```

1 .
2 .
3 //wenn sich ein User laenger als 10 Minuten im Shop bewegt
4 //wird das Session-Objekt gelöscht um es mit den aktuellsten Daten wieder zu
    füllen
5 if(isset($_SESSION['listeFavoriten'])){
6
7     //ueberprueft die Aktualitaet des Objektes (aufruf eine Funktion)
8     if(CheckAktualitaet(($time_to_akt_liste_favoriten*60),$_SESSION['
        listeFavoriten']->get_max_last_mod())){

```



```
9      //löscht das Session-Objekt
10      unset($_SESSION['listeFavoriten']);
11  }
12 }
13 .
14 .
```

Quelltext 4.16: Initialisierung der Datenaktualisierung für Favoritenliste

4.5 Übermittlung von Lagerbeständen

In diesem Abschnitt wird die Übermittlung der aktuellen Lagerbestände der einzelnen Filialen aus dem Warenwirtschaftssystem in den Web-Shop beschrieben. Wie im Abschnitt 3.1.3 auf Seite 23 erwähnt, können lediglich die Lagerbestände und somit die Verfügbarkeiten der Stammartikel aus dem Warenwirtschaftssystem ermittelt werden. Dies umfasst ca. 28.000 Artikel. Die Verfügbarkeiten der ca. 170.000 Preislistenartikel werden bei der Bildung der Mutter- und Mastertabelle eingetragen und bleiben für 24 Stunden fix. Die Ermittlung der Bestände aus dem Warenwirtschaftssystem und der Änderungsdaten sowie die Übertragung der Änderungsdaten und deren Import in die Web-Shop-Datenbank wird durch das Shell-Skript *bestandLager.sh* (siehe Quelltext im Anhang D.15 auf CD-ROM Seite 127) initialisiert. Der Aufruf dieses Skriptes erfolgt über die Jobsteuerung zu jeder vollen Stunde.

4.5.1 Ermittlung der Daten aus dem Warenwirtschaftssystem

Die Abfrage der Lagerbestände für die einzelnen Stammartikel wird in dem aufgerufenen PHP-Skript *lagerbestand_mention.php* (siehe Quelltext im Anhang D.16 auf CD-ROM Seite 128) durchgeführt. Darin wird

- jeder Stammartikel mit seiner Artikelnummer aus der Artikeltabelle im Intranet ermittelt.
- anhand der Artikelnummer das Lager mit dessen Bestand aus der Tabelle *AEL-LAGER* und *AEL* aus dem Warenwirtschaftssystem gelesen.
- eine CSV-Datei mit Artikelnummer, Bestand und der Verfügbarkeit erstellt.

Die in dem Skript erstellte CSV-Datei dient, als erste zu vergleichende Datei, der Ermittlung der Änderungsdaten zwischen Intranet und Warenwirtschaftssystem.

Ziel ist es, die Artikeltabelle im Intranet mit den aktuellen Lagerbeständen zu füllen. Dabei sollen lediglich die geänderten Daten aus dem Warenwirtschaftssystem aktualisiert werden. Bei der Erstellung der Änderungsdaten soll *diff* zum Einsatz kommen. Dazu werden zwei Textdateien benötigt, die miteinander verglichen werden können. Das PHP-Skript *lagerbestand_mention.php* erstellt eine Textdatei mit den aktuellen Lagerbeständen aus dem Warenwirtschaftssystem, die bei *diff* verwendet werden kann.

4.5.2 Ermittlung der Änderungsdaten

Die Änderungsdaten der Lagerbestände werden in zwei Schritten ermittelt. Zum einen wird eine Änderung zwischen dem Warenwirtschaftssystem und der Artikeltabelle im Intranet ermittelt und zum anderen die Änderung zwischen der Artikeltabelle im Intranet und der Artikeltabelle in der Web-Shop-Datenbank.

Um die Änderung zwischen Warenwirtschaftssystem und Intranet feststellen zu können, wird der Lagerbestand der Stammartikel aus der Artikeltabelle benötigt. Diese Lagerbestände werden, durch den Befehl in der *bestandLager.sh*

```
$mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'SELECT ArtNr,
hnr, Delitzsch, Halle, Hauptlager, Lieferbkt, Lieferbkt_Distri FROM
LagerBestand WHERE ArtNr>0' $mysqltestshopdb > $homepfad_M2S/
lagerbestand_nexus_stammartikel.txt
```

in die zweite zu vergleichende Textdatei geschrieben.

Anschließend werden diese beiden Dateien über *diff* verglichen und die daraus resultierenden ADD- bzw. REM-Dateien erstellt.

```
diff $homepfad_M2S/lagerbestand_nexus_stammartikel_sort.txt
$homepfad_M2S/lagerbestand_mention_sort.txt >
$homepfad_M2S/lagerbestand_mention_diff.txt
```

```
sed '/< /!d' $homepfad_M2S/lagerbestand_mention_diff.txt >
$homepfad_M2S/lagerbestand_mention_diff_rem.txt
```

```
sed '/> /!d' $homepfad_M2S/lagerbestand_mention_diff.txt >
```

```
$homepfad_M2S/lagerbestand_mention_diff_add.txt
```

Der Befehl,

```
$phppfad/php $phppfad_bestandLager/lagerbestand_nexus_update.php
```

ruft des PHP-Skript *lagerbestand_nexus_update.php* (siehe Quelltext im Anhang D.17 auf CD-ROM Seite 130), welches die Manipulation der Artikeltabelle im Intranet durchführt.

Ist die Artikeltabelle im Intranet mit den Lagerbeständen aktualisiert, kann diese mit der Artikeltabelle in Web-Shop-Datenbank verglichen werden. Mit den Befehlen

```
ssh prosoft@trinity "mysql -h $trinityhost -u $trinityuser -p  
$trinitypwd -e 'SELECT ArtNr,hnr,Delitzsch,Halle,Hauptlager,  
Lieferbkt FROM LagerBestand' $trinityshopdb" > lagerbestand_slave.txt
```

```
$mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e  
'SELECT ArtNr,hnr,Delitzsch,Halle,Hauptlager,Lieferbkt FROM LagerBestand'  
$mysqltestshopdb > lagerbestand_master.txt
```

werden die zwei zu vergleichenden Textdateien erstellt und anschließend mit dem Befehl

```
diff lagerbestand_slave_sort.txt lagerbestand_master_sort.txt >  
lagerbestand_diff.txt
```

verglichen.

4.5.3 Übertragung der Änderungsdaten in das Web-Shop-System

Die bei dem Vergleich von der Artikeltabelle im Web-Shop mit der Artikeltabelle im Intranet, entstehenden ADD- bzw. REM-Dateien werden mit den Befehlen

```
scp lagerbestand_diff_add.txt prosoft@trinity:/home/prosoft/tmp/  
lagerbestand_diff_add.txt
```

```
scp lagerbestand_diff_rem.txt prosoft@trinity:/home/prosoft/tmp/  
lagerbestand_diff_rem.txt
```

in das Web-Shop-System übertragen.

4.5.4 Import der Änderungsdaten in die Web-Shop-Datenbank

Der Befehl,

```
ssh prosoft@trinity "php /home/prosoft/bin/lagerbestand_update.php"
```

in der *bestandLager.sh* importiert die Änderungsdaten in die Artikeltabelle der Web-Shop-Datenbank. Dabei werden die übertragenen ADD- bzw. REM-Dateien durch das aufgerufene PHP-Skript *lagerbestand_update.php* (siehe Quelltext im Anhang D.18 auf CD-ROM Seite 131) eingelesen und anhand derer die Manipulation der Artikeltabelle vorgenommen.

4.5.5 Anzeige der Daten im Web-Shop

Die Anzeige der Verfügbarkeiten wird in zwei Kategorien unterteilt. Zum einen werden die Lagerbestände der Stammartikel in den Filialen angezeigt und zum anderen die Verfügbarkeiten der Preislistenartikel. Die Verfügbarkeiten der Preislistenartikel werden zur Laufzeit bei der Erstellung der Artikeltabelle aus der Web-Shop-Datenbank ausgelesen.

Für die Lagerbestände der Stammartikel wird in der *index.php* ein Session-Objekt erzeugt und mit der Artikelnummer, den Beständen in den einzelnen Filialen sowie der Zeit der letzten Aktualisierung gefüllt.

```
1 //Nur einmal bis Session-Objekt bestandLager angelegt ist (wird beim abmelden in  
  abmelden.php geleert)  
2 if(!isset($_SESSION['bestandLager'])){  
3     //setzt anhand des Lager-Arrays den Select zusammen  
4     if(count($lager_arr)>0){  
5         $min_bestand="";  
6  
7         $i=0;  
8
```

```

9      foreach($lager_arr as $key => $lager_name){
10          $suche_nach_lager_name.="L.$lager_name, ";
11
12          if($i==0){
13              $min_bestand.=$lager_name.">0";
14          }
15          else{
16              $min_bestand.="_OR_". $lager_name.">0";
17          }
18          $i++;
19      }
20      $min_bestand.=")";
21  }
22
23  //Abfrage der LagerBestandTabelle nach ArtNr, Beständen und Benennungen
24  $query_bestandLager = "SELECT_L.last_mod, L.ArtNr, ".$suche_nach_lager_name.
        "A.Benennung, A.bez_itscope, A.itscope_FROM_ $lagerbestandtable_as_L,
        $artikeltable_as_A_WHERE_ $min_bestand_AND_A.ArtNr=L.ArtNr_AND_L.hnr=A.
        hnr_ORDER_BY_L.last_mod_DESC";
25
26  $result_bestandLager = mysql_query($query_bestandLager, $dbh1);
27
28  //wenn Abfrage mindestens ein Ergebnis liefert Objekt erstellen
29  if(mysql_num_rows($result_bestandLager)!=0){
30      //erstellt ein SESSION-Objekt mit dem jüngsten letzten Aenderungsdatum
        für alle Artikel ohne Lagerbestand
31      $_SESSION['bestandLager'] = new bestandLager(mysql_result(mysql_query("
        SELECT_max(last_mod)_FROM_ $lagerbestandtable", $dbh1), 0, 0));
32
33      while($row = mysql_fetch_array($result_bestandLager)){
34          //fügt die ArtNr einem Session-Objekt-Array hinzu
35          $_SESSION['bestandLager']->add_ArtNr($row['ArtNr']);
36          //fügt die letzte Aenderung (last_mod) einem Session-Objekt-Array
        hinzu mit Index-Schlüssel ArtNr
37          $_SESSION['bestandLager']->add_last_mod($row['ArtNr'], $row['last_mod']
        ');
38
39          $itscope=$row['itscope'];
40
41          //fügt die Benennung des Artikels einem Session-Objekt-Array hinzu
42          if($itscope==TRUE){
43              $_SESSION['bestandLager']->add_Benennung($row['ArtNr'], $row['
        bez_itscope']);
44          }
45          else{

```

```

46         $_SESSION['bestandLager']->add_Benennung($row['ArtNr'], $row['
           Benennung']);
47     }
48 }
49
50 if(count($lager_arr)>0){
51     foreach($lager_arr as $key => $lager_name){
52         //bewegt den internen Datensatz-Zeiger eines Anfrageergebnisses
           zum Datensatz mit der übergebenen Nummer
53         //hier wieder auf fetch_array index=0
54         mysql_data_seek($result_bestandLager, 0);
55
56         while($row = mysql_fetch_array($result_bestandLager)){
57             //fügt einem mehrdimensionalen Assoziativen Array jedem
           Lager die Verfügbarkeit hinzu mit Index-Schlüssel ArtNr
58             $_SESSION['bestandLager']->add_bestand($lager_name, $row['
           ArtNr'], $row[$lager_name]);
59         }
60     }
61 }
62 }
63 }

```

Quelltext 4.17: Auszug der *index.php* für die Funktion *Lagerbestand*

Zur Visualisierung der Verfügbarkeiten werden vier Symbole mit unterschiedlicher Farbgebung kreiert. Die Farbe grün bedeutet "ab Lager", grün-gelb "ab Lager bei Lieferanten", gelb "circa 10 Werkzeuge" oder "Verfügbarkeit kurzfristig innerhalb 1 Werktag" und die Farbe rot "auf Anfrage". Dabei wird grün nur für die Anzeige der Bestände in den Filialen bei den Stammartikel verwendet. Die Farbe gelb wird zwischen Stammartikel und Preislistenartikel differenziert. In beiden Fällen hat gelb eine andere Benennung. Diese Benennung kann wie folgt beschrieben werden:

grün (ab Lager) Der Artikel hat in dieser Filiale einen Bestand von mindesten eins.

grün-gelb (ab Lager bei Lieferanten) Der Artikel ist bei mindestens einem Lieferanten auf Lager.

gelb (Verfügbarkeit kurzfristig innerhalb 1 Werktag) Der Artikel ist in dieser Filiale nicht auf Lager, kann jedoch innerhalb eines Werktages aus einer anderen Filiale geordert werden.

gelb (circa 10 Werkzeuge) Die Lieferzeit dieses Artikels ist bei dem Lieferanten mit kleiner gleich zehn Tagen angegeben.

rot (auf Anfrage) Die Lieferzeit des Artikels ist bei dem Lieferanten größer als zehn Tage oder bei diesem nicht verfügbar.

Die Symbole werden in der Artikelliste bzw. der Detailansicht angezeigt. Dazu wird bei der Generierung der Artikelliste oder der Detailansicht die Verfügbarkeit für jeden darzustellenden Artikel aus der Artikeltabelle abgefragt. Die implementierte Funktion *Verfuegbarkeit()* (siehe Quelltext im Anhang D.19 auf CD-ROM Seite 132) setzt je nach Verfügbarkeit das entsprechende Symbol (siehe Abbildung 4.10 roter Pfeil) in der Artikelliste.

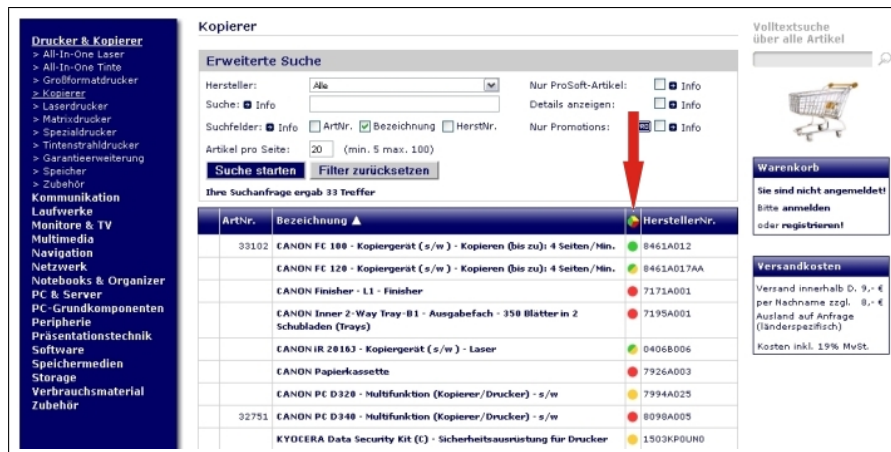


Abbildung 4.10: Artikelliste mit Verfügbarkeitsymbolen

Die Symbole bieten eine Linkfunktion. Somit kann der Benutzer diese Symbole anklicken und es öffnet sich ein Pop-Up-Fenster. Über diesen Link wird die Artikelnummer eines Stammartikels als GET-Variable dem inkludierten PHP-Skript *lager_bestand.php* (siehe Quelltext im Anhang D.20 auf CD-ROM Seite 134) übergeben, welches für den Inhalt in dem Pop-Up-Fenster zuständig ist.

Darstellung der Lagerbestände

Das PHP-Skript *lager_bestand.php* realisiert die unterschiedliche Darstellung der Verfügbarkeiten bzw. Lagerbestände eines Artikels. Dieses Skript überprüft, ob die übergebene Artikelnummer in dem Session-Objekt der Lagerbestände enthalten ist. Ist dies der Fall, wird mit Hilfe der Funktion *GetLagerBestand* (siehe Quelltext im Anhang D.21 auf CD-ROM Seite 135) das Datum (siehe Abbildung 4.11 auf Seite 74 grüner Pfeil) und die Uhrzeit der letzten Aktualisierung (siehe Abbildung 4.11 auf Seite 74 blauer Pfeil) sowie die Benennung des Artikels (siehe Abbildung 4.11 auf Seite 74 roter Pfeil) aus dem Session-Objekt ermittelt. Weiterhin werden mit der Funktion *GetLagerBestandFiliale()*

(siehe Quelltext im Anhang D.22 auf CD-ROM Seite 135) die einzelnen Bestände der Filialen aus dem Session-Objekt gelesen.

Für die Darstellung des Pop-Up-Fensters ist die Template-Datei *lager_bestand.html* (siehe Quelltext im Anhang D.23 auf CD-ROM Seite 135) zuständig. Damit ist es möglich, verschieden Inhalte wiederzugeben. Zum einen die Lagerbestände für einen Stammartikel (siehe Abbildung 4.11) und zum anderen die Erläuterung zu den Verfügbarkeiten der Preislistenartikel (siehe Abbildung 4.12 rote Pfeile).



Abbildung 4.11: Lagerbestände für einen Stammartikel



Abbildung 4.12: Erläuterung zu den Verfügbarkeiten der Preislistenartikel

Mit dieser beschriebenen Funktion werden die Verfügbarkeiten in der Artikelliste und der Detailansicht dem Benutzer des Web-Shops visuell dargestellt. Über einen Klick stehen im darüber hinaus eine detaillierte Erläuterung der unterschiedlichen Verfügbarkeiten als auch eine genaue Auflistung der Bestände in den einzelnen Filialen der ProSoft Krippner GmbH zur Verfügung.

Aktualisierung der Daten während einer Benutzersitzung

Für eine Aktualisierung der Lagerbestände, während einer Sitzung des Benutzers im Web-Shop, wird in der *index.php* (Siehe Quelltextauszug 4.18) die Funktion *CheckAktualitaet()* (siehe Quelltext im Anhang D.7 auf CD-ROM Seite 114) aufgerufen. Sind die Bedingungen in der Funktion erfüllt, wird das Session-Objekt mit den Lagerbeständen gelöscht und anschließend, mit den aktuellen Beständen aus der Web-Shop-Datenbank, neu erstellt.

```
1 .
2 .
3 //wenn sich ein User laenger als 1 Stunde im Shop bewegt
4 //wird das Session-Objekt gelöscht um es mit den aktuellsten Daten wieder zu
   fuellen
5 if(isset($_SESSION['bestandLager'])) {
6
7     //ueberprueft die Aktualitaet des Objektes (aufruf eine Funktion)
8     if(CheckAktualitaet(($time_to_akt_bestand_lager*60),$_SESSION['bestandLager '
       ]->get_max_last_mod())) {
9         //löscht das Session-Objekt
10        unset($_SESSION['bestandLager']);
11    }
12 }
13 .
14 .
```

Quelltext 4.18: Initialisierung der Datenaktualisierung für *Lagerbestand*

4.6 Import von Vorgängen aus dem Web-Shop in das Warenwirtschaftssystem

Dieser Abschnitt befasst sich mit der Umsetzung des Imports von Angeboten und Bestellungen aus dem Web-Shop in das Warenwirtschaftssystem.

4.6.1 Ermittlung der Daten im Web-Shop

Damit ein Angebot oder eine Bestellung in das Warenwirtschaftssystem importiert werden kann, müssen im Vorfeld alle benötigten Daten (siehe Abschnitt 3.3.2 auf Seite 33)

im Web-Shop gesammelt werden.

Die benötigten Daten, für einen eingeloggtten Benutzer, werden während einer Sitzung in einem Session-Objekt gesammelt. Dieses Session-Objekt wird nur erstellt, wenn der Benutzer für diese Funktion freigeschaltet ist. Die Überprüfung, ob die Funktion einem Benutzer zugewiesen ist, findet in der *makeID.php* (siehe Quelltextauszug 4.19) statt.

```
1 .
2 .
3 .
4 // Abfrage der User-Daten
5 $res=mysql_query("SELECT _UserID , _Name, _Vorname, _Preistyp , _Nummer, _AlleArtikel , _
    AnzPS, _MentionKundennummer, _MentionBez , _atAusgang , _Favoritenliste , _
    OnlineBestellung , _Anfrage , _KeinBestellen _FROM_ $kundentable _WHERE_ ( Login='
    $Login ' ) _AND_ ( Passwort=MD5( ' $Passwort ' ) )" , $dbh1 );
6 .
7 .
8 .
9 //ob der eingeloggte User eine aus Mention gesteuerte Favoritenliste sieht
10 $_SESSION[ 'hat _OnlineBestellung ']=mysql_result( $res,0,11) ;
11 .
12 .
13 .
```

Quelltext 4.19: Überprüfung der Logindaten und der Onlinebestellung-Funktionalität

Eine Anfrage für ein Angebot (siehe Abbildung 4.13 auf Seite 77 roter Pfeil) oder eine Bestellung (siehe Abbildung 4.13 auf Seite 77 grüner Pfeil) ist nur über die Warenkorbseite möglich.

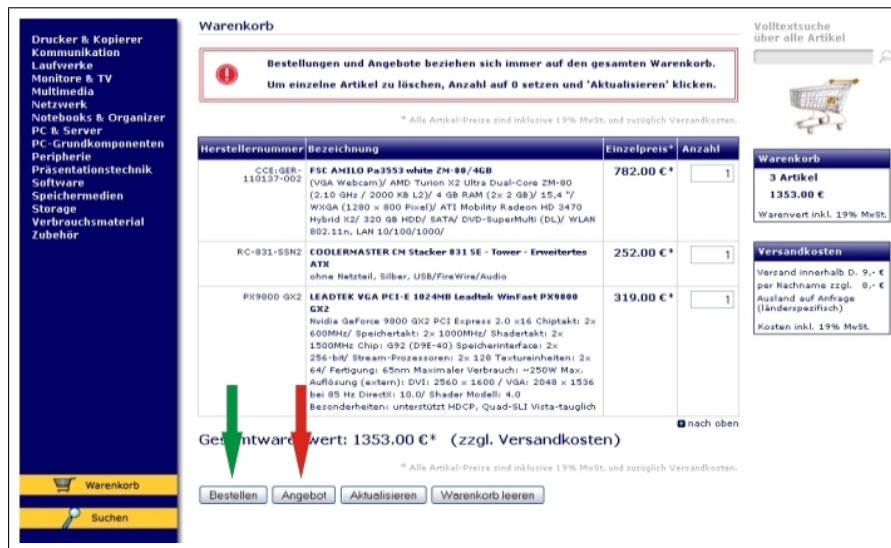


Abbildung 4.13: Die Warenkorbansicht im Web-Shop

Somit wird das Session-Objekt erstellt, wenn sich der Benutzer seinen Warenkorb ansieht. Der Warenkorb wird mit dem PHP-Skript *warenkorb.php* realisiert. Zu Beginn wird die Freischaltung für eine Onlinebestellung überprüft und das Session-Objekt angelegt (siehe Quelltextauszug 4.20).

```

1 //wenn der Kunde für OnlineBestellung freigeschaltet ist (nur für diese Kunden
  ist der Online Bestellvorgang von Bedeutung)
2 if($_SESSION['hat_OnlineBestellung']==1){
3   //erstellt das Session-Objekt onlineBestellung NUR 1 mal
4   if(!isset($_SESSION['onlineBestellung'])){
5     $_SESSION['onlineBestellung'] = new onlineBestellung($_SESSION['
      MentionKdnr']);
6   }
7 }

```

Quelltext 4.20: Erstellen des Session-Objekts für *Onlinebestellung*

Der Benutzer hat in dem Warenkorb die Möglichkeit, die darin enthaltenen Positionen zu bestellen oder sich ein Angebot erstellen zu lassen. In beiden Fällen werden alle notwendigen Daten für die Onlinebestellung gesammelt und in das Session-Objekt eingetragen (siehe Quelltext im Anhang D.24 auf CD-ROM Seite 136). Es werden dabei folgende Daten ermittelt:

- die Anzahl an Positionen
- die Herstellernummer jeder Position

- die Stückzahl jeder Position
- der Brutto- und der Nettopreis jeder Position

4.6.2 Aktualisierung der Daten während einer Benutzersitzung

Eine Aktualisierung der Daten ist nur dann notwendig, wenn neue Positionen über die Artikeltabelle oder der Detailansicht in den Warenkorb aufgenommen werden, die Stückzahl einer Position geändert wird, eine Position aus dem Warenkorb entfernt oder der gesamte Warenkorb gelöscht wird.

Damit die Daten in das Session-Objekt neu eingelesen werden, wird das Session-Objekt in den beschriebenen Fällen gelöscht (siehe Quelltextauszug 4.21 auf Seite 78).

```
1 .
2 .
3 .
4 if($aktualisieren!=""){
5 .
6 .
7 .
8     if(($auswahl)&&($stueck!=$auswahl[$k])){
9         mysql_query("UPDATE_ $warenktable_SET_Stueck='$auswahl[$k]' _WHERE_ ( hnr='
            $hnr ' ) _AND_ ( UserID='$_SESSION[ UserID ] ' ) ", $dbh1);
10    }
11    if(($auswahl)&&($auswahl[$k]==0)){
12        mysql_query("DELETE_FROM_ $warenktable _WHERE_ ( UserID='$_SESSION[ UserID ] ' )
            _AND_ ( Stueck='0 ' ) ", $dbh1);
13    }
14
15    //löscht das Session-Objekt onlineBestellung
16    unset($_SESSION['onlineBestellung']);
17 }
18
19 if($leeren!=""){
20     mysql_query("DELETE_FROM_ $warenktable _WHERE_ ( UserID='$_SESSION[ UserID ] ' ) ",
        $dbh1);
21
22    //löscht das Session-Objekt onlineBestellung
23    unset($_SESSION['onlineBestellung']);
24 }
```

```
25
26 //löschen der Artikel im Warenkorb welche nicht im Shop enthalten sind EOL
27 if ($action=="Entfernen"){
28
29     mysql_query("DELETE FROM $warenktable WHERE ( UserID='$_SESSION[ UserID ] ' ) AND
        _ ( hnr='$hnr_fehlt ' ) ", $dbh1);
30
31     //löscht das Session-Objekt onlineBestellung
32     unset($_SESSION['onlineBestellung']);
33 }
34 .
35 .
36 .
37 //Wenn auf "In den Korb" geklickt wurde
38 if ($in_den_korb==1){
39 .
40 .
41 .
42     //löscht das Session-Objekt onlineBestellung
43     unset($_SESSION['onlineBestellung']);
44 }
45 .
46 .
47 .
```

Quelltext 4.21: Löschen des Session-Objekte für *Onlinebestellung*

4.6.3 Erstellen einer Bestellung oder eines Angebotes im Web-Shop

Während des Bestellvorgangs hat der Benutzer die Möglichkeit, neben der Bezahlung die Art des Versandes anzugeben. Dies ist daher von Interesse, da bei Versand über DHL (siehe Abbildung 4.14 grüner Pfeil) als auch bei Versand per Nachnahme (siehe Abbildung 4.14 roter Pfeil) zusätzliche Kosten entstehen. Diese Kosten müssen in der Bestellung als Positionen eingetragen werden, damit diese nach dem Import im Warenwirtschaftssystem vorhanden sind.

Abbildung 4.14: Auswahl der Bezahlungsart im Bestellvorgang

Hat der Benutzer, im nächsten Schritt des Bestellvorgangs, die AGB der Firma Pro-Soft Krippner akzeptiert (siehe Abbildung 4.15 auf Seite 81 grüner Pfeil) und die Bestellung über den Schalter "Absenden" abgeschlossen (siehe Abbildung 4.15 auf Seite 81 roter Pfeil), wird das PHP-Skript *online_bestellung.php* (siehe Quelltext im Anhang D.25 auf CD-ROM Seite 139) aufgerufen.

allgemeine Geschäftsbedingungen

AGB drucken

ALLGEMEINE GESCHÄFTSBEDINGUNGEN (Stand Juni 2008)

§ 1 Allgemeines, Geltungsbereich

Die folgenden Allgemeinen Geschäftsbedingungen (AGB) regeln das Vertragsverhältnis zwischen der ProSoft Krippner GmbH, Löhstraße 25, 04105 Leipzig (im Folgenden ProSoft genannt) einerseits und den Verbrauchern und Unternehmern, die das Internetangebot von ProSoft nutzen (im Folgenden "Käufer" genannt). Die AGB betreffen die Nutzung der Website www.prosoft-krippner.com/shop sowie alle zu dieser Domain gehörenden Subdomains. Maßgeblich ist jeweils die zum Zeitpunkt des Vertragsschlusses gültige Fassung. Die Vertragssprache ist Deutsch.

Diese Vertragsbedingungen geltend ausschließlich; entgegenstehende oder von unseren Vertragsbedingungen abweichende Bedingungen des Kunden erkennen wir nicht an, es sei denn, wir hätten ausdrücklich schriftlich ihrer Geltung zugestimmt. Unsere Vertragsbedingungen gelten auch dann, wenn wir in Kenntnis entgegenstehender oder von unseren Bedingungen abweichender Bedingungen des Kunden die vertraglichen Leistungen erbringen.

Verbraucher im Sinne dieser Geschäftsbedingungen sind natürliche Personen, die mit ProSoft in Geschäftsbeziehung treten, ohne dass dies ihrer gewerblichen oder selbständigen beruflichen Tätigkeit zugerechnet werden kann.

Unternehmer im Sinne dieser Geschäftsbedingungen sind natürliche und juristische Personen oder rechtsfähige Personengesellschaften, die in Ausübung ihrer gewerblichen oder selbständigen beruflichen Tätigkeit mit ProSoft in eine Geschäftsbeziehung treten.

☒ * Der Kauf der Ware erfolgt auf der Grundlage der **allgemeinen Geschäftsbedingungen (AGB)** des Betreibers des Onlineshops. Der Käufer bestätigt, dass er die genannten **AGB** gelesen und ausgedruckt hat und sich mit der Einbeziehung der **AGB** in den Kaufvertrag einverstanden erklärt.

☐ Warenkorb nach dem Absenden der Bestellung löschen?

Absenden

Die mit einem Stern (*) gekennzeichneten Felder sind Pflichtfelder!
Bitte verwenden Sie keine Sonderzeichen und Umlaute.

Abbildung 4.15: Abschließen des Bestellvorgangs

Das PHP-Script *online_bestellung.php* ermittelt alle Daten, die für die Bildung der Hilfstabelle in der Web-Shop-Datenbank notwendig sind. Dies umfasst Daten, die nicht mit die einzelnen Positionen aus dem Warenkorb verknüpft sind. Das sind unter anderem

- der Typ der Bestellung/Angebot ('A','B')
- der Brutto- und Nettopreis bei evtl. Versandkosten
- der Brutto- und Nettopreis bei evtl. Nachnahmegebühr
- die generierte eindeutige ID der Bestellung/Angebot
- der Kommentartext zu der Bestellung/Angebot
- die Uhrzeit der Bestellung/Angebot
- das Datum der Bestellung/Angebot

Des Weiteren wird durch das PHP-Skript eine E-Mail generiert, die dem eingetragenen Benutzer des Benutzers zugesendet wird. Darin ist die Information enthalten, dass eine Onlinebestellung in das Warenwirtschaftssystem eingegangen ist. Im Folgenden

muss diese Onlinebestellung oder Angebot in eine interne Bestellung bzw. Angebot umgewandelt werden.

Sind alle benötigten Daten für die Hilfstabelle vorhanden, können diese eingetragen werden. Die Hilfstabelle sammelt alle ausgelösten Bestellungen oder Angebote aus dem Web-Shop, bis die Jobsteuerung die Übertragung dieser Daten in das Intranet initialisiert.

4.6.4 Übertragung der Daten in das Warenwirtschaftssystem

Damit die Bestellungen in dem Warenwirtschaftssystem bearbeitet werden können, müssen die Daten aus dem Web-Shop in das Intranet übermittelt werden. Dafür wird über die Jobsteuerung das Shell-Skript *importBestellung.sh* (siehe Quelltext im Anhang D.26 auf CD-ROM Seite 144) im Intranet aufgerufen.

Die Übertragung der Daten wird in dem Shell-Skript *copy_OnlineBestellung.sh* (siehe Quelltext im Anhang D.27 auf CD-ROM Seite 144) durchgeführt. In dem Fall der Onlinebestellungen kann auf eine Übertragung der Änderungsdaten verzichtet werden, da nach Abschluss der Übertragung die Hilfstabelle im Web-Shop sowie die Hilfstabelle im Intranet geleert werden und somit keine Änderung ermittelt werden kann.

Ein SQL-Befehl ermittelt über eine SSH-Verbindung alle Daten aus der Web-Shop-Datenbank und speichert diese in eine CSV-Datei. Direkt im Anschluss wird die Datenbanktabelle im Web-Shop geleert, damit neue Onlinebestellungen in diese aufgenommen werden können. Über *scp* wird die erstellte CSV-Datei in das Intranet-System kopiert. Der Befehl

```
$mysqlpfad/mysqlimport --local -h $mysqlhost -u $mysqluser -p$mysqlpwd  
--fields-enclosed-by='#' --fields-terminated-by=';' --  
--lines-terminated-by='\n' --columns bestellid,oskundennr,hnr,aridnr,  
osbeltyp,oszeit,osdatum,oszusatz,opvkpreis,opstueck,opvkbrutt,Name  
$mysqltestshopdb $homepfad_M2S/OnlineBestellung.txt
```

importiert die Daten aus der CSV-Datei in die Hilfstabelle im Intranet.

Alle notwendigen Daten befinden sich im Intranet-System, können aber noch nicht in der Warenwirtschaft verwendet werden. Das Shell-Skript *import_OnlineBestellung.sh* (siehe Quelltext im Anhang D.28 auf CD-ROM Seite 145) realisiert den Datenimport.

4.6.5 Import der Bestellung/Angebot in das Warenwirtschaftssystem

Die Daten aus der Hilfstabelle müssen so in das Warenwirtschaftssystem importiert werden, dass sie durch die bereits bestehenden, System internen, Routinen genutzt werden können. Dazu müssen die Daten in bestimmte Tabelle und Felder geschrieben werden. Der Import der Daten wird in dem implementierten PHP-Skript *get_online_bestellung.php* (siehe Quelltext im Anhang D.29 auf CD-ROM Seite 145) durchgeführt.

Das PHP-Skript besitzt dabei folgende Funktionalitäten:

- Ermittlung der internen Artikelnummer eines Stammartikels
- Ermittlung der Bestellnummer und des Distributors eines Preislistenartikels
- Berechnung des gesamten Nettopreises der Bestellung eines Kunden
- Berechnung des gesamten Bruttopreises der Bestellung eines Kunden
- Berechnung der gesamten Mehrwertsteuer der Bestellung eines Kunden
- Eintragen eines Bestellkopfdatensatzes in die Tabelle OBESTKK
- Ermittlung der maximalen OBESTKK-ID
- Eintragen der Bestellpositionendatensätze für jede Bestellung in die Tabelle OBESTKP
- Versenden einer E-Mail für Artikel die nicht aus dem System ermittelt werden können

Die Ermittlung der internen Artikelnummer für Stammartikel bzw. die Bestellnummer und der Distributor für einen Preislistenartikel ist von daher notwendig, da dies die Hauptreferenzen im Warenwirtschaftssystem sind. Diese Daten sind im Web-Shop nicht vorhanden und müssen vor dem Import, anhand der Herstellernummer ermittelt werden.

Jede einzutragene Position in OBESTKP muss einer Bestellung in OBESTKK zugeordnet werden. Die Verbindung dieser Datensätze ist das Feld *OSLFDNUM* in der OBESTKK. Die maximale *OSLFDNUM* muss vor dem Eintragen einer Position ermittelt werden.

Sind alle Daten aus der Hilfstabelle im Intranet in das Warenwirtschaftssystem importiert, stehen die Daten für eine weitere Bearbeitung zur Verfügung. Die Hilfstabelle im Intranet wird am Ende des Shell-Skripts geleert.

4.6.6 Angebot und Bestellung im Warenwirtschaftssystem

Damit die Bestellungen im Warenwirtschaftssystem bearbeitet werden können, müssen die Onlinebestellungen, wie im Abschnitt (4.6.3 auf Seite 79) erwähnt, in interne Bestellungen umgewandelt werden. Dies erfolgt über den Menüpunkt "Online Bestellungen" (siehe Abbildung 4.16 auf Seite 84 roter Pfeil) im "mention".

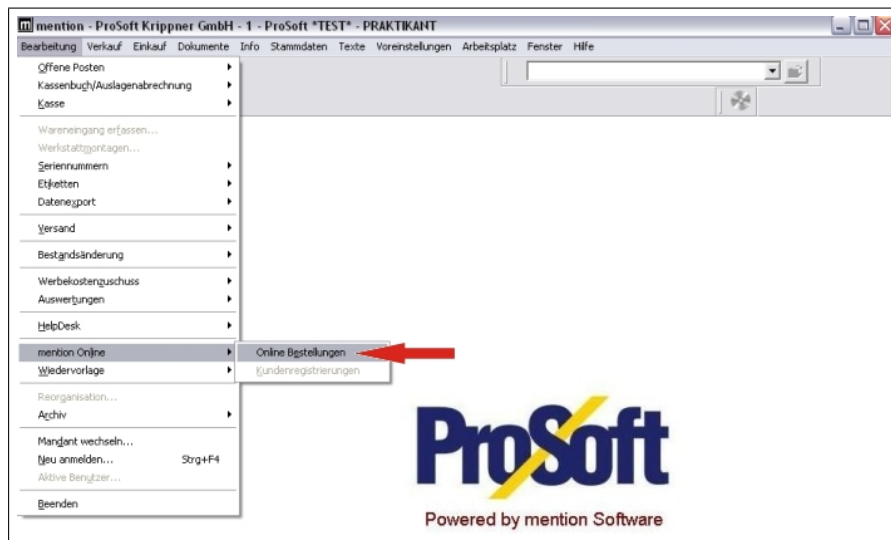


Abbildung 4.16: Menüpunkt im "mention" zur Umwandlung der Onlinebestellungen in interne Bestellungen

Der sich öffnende Programmteil erlaubt das Suchen nach Belegen, die in der Datenbanktabelle OBESTTKK bzw. OBETSKP eingetragen sind. Über den Schalter "Suchen" (siehe Abbildung 4.17 auf Seite 85 roter Pfeil) werden alle eingetragenen Belege mit ihren Positionen angezeigt (siehe Abbildung 4.17 auf Seite 85 grüne Pfeile). Dabei werden die Daten Uhrzeit und Datum der Bestellung, Kundennummer, Belegtyp, gesamter Nettopreis der Bestellung, Stückzahl der einzelnen Positionen sowie Einzelpreis und Gesamtpreis der Positionen ausgelesen und angezeigt (siehe Abbildung 4.17 auf Seite 85 blaue Pfeile). Die Umwandlung in interne Bestellungen bzw. Angebote erfolgt über den Schalter "Übernahme" (siehe Abbildung 4.17 auf Seite 85 schwarzer Pfeil.)

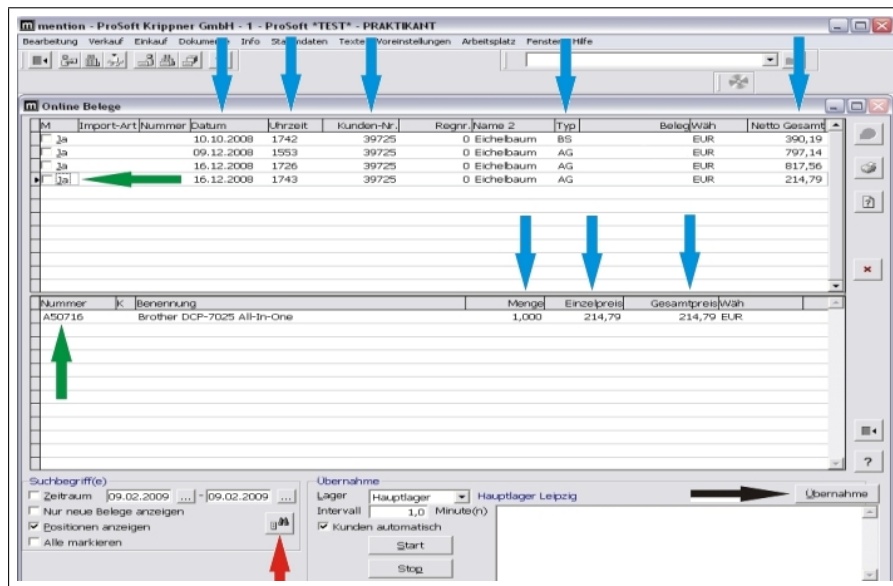


Abbildung 4.17: Onlinebestellungen vor der Umwandlung in eine interne Bestellungen

Die Umwandlung der Onlinebestellung in eine interne Bestellung wird in dem Fenster (siehe Abbildung 4.18 roter Pfeil) bestätigt. Zusätzlich enthält nach der Übernahme jede Bestellung eine interne Belegnummer (siehe Abbildung 4.18 grüner Pfeil). Somit ist der Import einer Bestellung oder eines Angebots aus dem Web-Shop in das Warenwirtschaftssystem abgeschlossen.

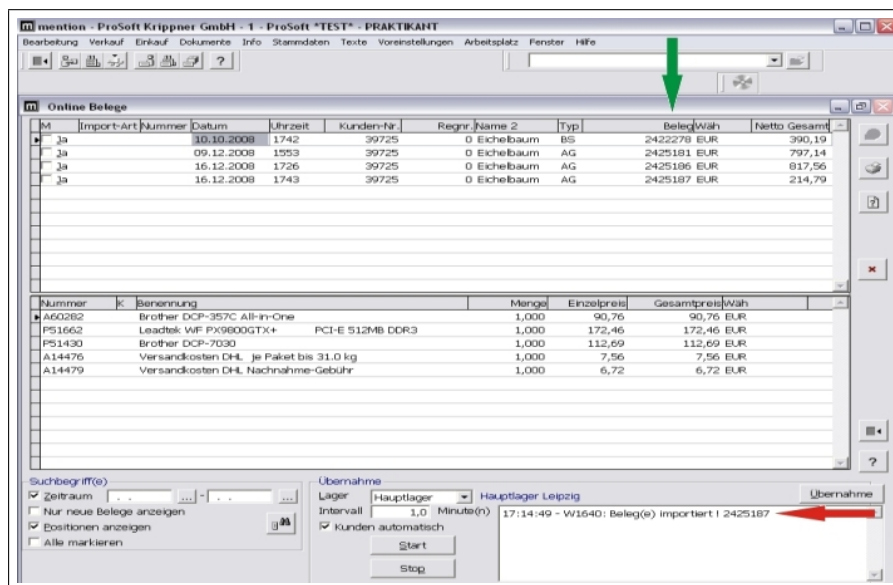


Abbildung 4.18: Onlinebestellungen nach der Umwandlung in eine interne Bestellungen

4.7 Zusammenfassung

In dem Kapitel Realisierung wurde die Umsetzung der Zielstellungen durchgeführt. Dabei wurde die Jobsteuerung für die automatische Initialisierung des Datenabgleiches zwischen Warenwirtschaftssystem und Web-Shop beschrieben und an einem praktischen Beispiel vollzogen. Weiterhin wurde für jede der vier umzusetzenden Funktionalitäten die Ermittlung der Daten aus dem Warenwirtschaftssystem bzw. dem Web-Shop, die Ermittlung der Änderungsdaten, deren Import in die entsprechenden Datenbanktabellen sowie die Anzeige dieser Daten im Web-Shop implementiert und dokumentiert.

5 Resümee

In dieser Bachelorarbeit wurde ein Konzept zur Entwicklung eines bidirektionalen Datenflusses zwischen einem Warenwirtschaftssystem und einem Web-Shop sowie dessen Umsetzung vorgestellt. Die im Vorfeld getätigten Überlegungen befassten sich sowohl mit der IT-Infrastruktur der Firma ProSoft Krippner als auch mit den im Web-Shop eingesetzten Technologien. Weiterhin wurden verschiedene Grundlagen in den Bereichen der Warenwirtschaft sowie bei der Entwicklung von dynamischen Webseiten mit Hilfe von PHP, MySQL und Templates vermittelt.

Im weiteren Verlauf der Arbeit wurden die Anforderungen des umzusetzenden Datenflusses herausgearbeitet. Die Analyse der Datenhaltung in den beiden Systemen war ebenso von Bedeutung, wie die Überlegungen über eine performante Integration der Daten in den Web-Shop und deren nutzerfreundliche Anzeige. Dabei spielte die Lastensenkung im Intranet eine wichtige Rolle. Dies wurde durch die Ermittlung und Übertragung von Änderungsdaten erreicht. Bei der Beschreibung der technischen Umsetzung wurde auf die Ermittlung der Daten aus den beiden Systemen, die Bereitstellung der Änderungsdaten und deren Übertragung eingegangen.

Abschließend kann gesagt werden, dass die umzusetzenden Funktionen für den Web-Shop durch diese Arbeit erreicht wurden und sich diese zur Zeit im produktiven Einsatz befinden. Somit stehen den Benutzer des Web-Shops sowie den Mitarbeitern der Firma ProSoft Krippner sinnvolle Erweiterungen zur Verfügung, welche einen besseren Austausch an Informationen sowie einen verbesserten Arbeitsfluss ermöglichen.

5.1 Ausblicke

Vorausschauend kann man den umgesetzten bidirektionalen Datenfluss zwischen Web-Shop und Warenwirtschaftssystem für weitere nützliche Funktionalitäten im Web-Shop einsetzen. Somit ist es zum Beispiel denkbar, dass man den Benutzern des Web-Shops eine Verfolgung ihrer Bestellungen, vom Bestelleingang über Bestellstatus bis hin zur Anzeige der Versendung der Bestellung bereitstellen kann. Allgemeiner kann gesagt werden, dass nahezu alle Daten aus dem Warenwirtschaftssystem in den Web-Shop

importiert werden können.

Das Thema der Datenübertragung kann weiterhin durch eine Umsetzung mit Hilfe von XML neu aufgegriffen werden. Mit XML ist es möglich, Daten in einem unabhängigen Format zu beschreiben und zu speichern. Dafür müsste eine XML kompatible Schnittstelle in beiden Systemen vorhanden sein bzw. diese implementiert werden. Ist diese Schnittstelle in beiden Systemen vorhanden, können die ermittelten Daten aus dem jeweiligen System mit Hilfe von XML-Dokumenten in das andere System übertragen werden. Für eine Integration der somit übertragenen Daten in den Web-Shop bietet PHP in der Version fünf ebenfalls eine geeignete Möglichkeit an. Mit PHP ist es möglich, XML-Dokumente zu parsen und somit die übermittelten Daten auszulesen. Die Übertragung der Daten mit XML ist ein sehr spannendes und zukunftsicheres Verfahren, welches den Anlass bietet, sich über eine Umsetzung mit Hilfe dieser Technologie auseinander zu setzen.

In dieser Arbeit wurde auf dieses Verfahren nicht näher eingegangen, da das verwendete Warenwirtschaftssystem keine Möglichkeit bietet, die darin enthaltenen Daten in XML-Dokumenten bereitzustellen bzw. XML-Dokumente verarbeiten zu können.

A Fragenkatalog

Kunde:

Allgemein:

Welche Web-Shops nutzen Sie?

Nutzen Sie Shops zur Information oder zum Kauf?

Welche Vor- und Nachteile hat unser Shop gegenüber den anderen Shops?

Was müsste sich unbedingt verbessern, damit Sie unseren Shop stärker nutzen würden?

Volle Abwicklung ohne Anmeldung/Login?

Anzeige von Angeboten aus der Warenwirtschaft im Web-Shop?

Produktdarstellung und -suche:

Produkte finden, Suchfunktion, Beschreibungen, Bilder, PDF?

Struktur der Produktgruppen verständlich (Menü-Baum)?

Favoritenliste?

Liste der Artikel aussagekräftig (Details, Bestellbutton in der Liste)?

Filterfunktionen (Liste einschränken, Suche nach Kriterien, Topseller)?

Zubehör zum Artikel / Basisartikel zum Zubehör / Verwandte Produkte?

Warenkorb (verständlich, Druckansicht, Auftragsverfolgung/Status)?

Lagerbestand (Aktualisierung, Filialen)?

Produkte vergleichen?

B Spezifizierung der Datenbanktabellen

Datenbanktabelle "*Muttertabelle*"

Tabelle B.1: Datenbanktabelle "*Muttertabelle*"

Name	Beschreibung	Datentyp	Länge
dis	Distributorname	VARCHAR	50
lnr	Lieferantennummer	INT	11
hnr	Herstellerartikelnummer	VARCHAR	30
anr	Artikelnummer	VARCHAR	30
ean	EAN	BIGINT	13
bez	Bezeichnung	VARCHAR	100
preis	EK Preis aus den Preislisten	DECIMAL	15,2
beschrei	erweiterte Benennung	TEXT	
lp	Listenpreis	DECIMAL	15,2
uvp	Unverbindliche Preisempfehlung	DECIMAL	15,2
gr1	Gruppe1	VARCHAR	30
gr2	Gruppe2	VARCHAR	30
gr3	Gruppe3	VARCHAR	30
psgrid	ID für ProSoft Gruppe	INT	11
psgr2	ProSoft Gruppe	VARCHAR	30
lager	Lagerbestand	VARCHAR	30
lfz	Lieferzeit	VARCHAR	30
hst	Hersteller	VARCHAR	30
hstid	ID für Hersteller	INT	11
eol	EOL-Kennzeichnung eines Artikels	VARCHAR	10
ausbl	ob ein Artikel aus dem Shop ausgeblendet werden soll	VARCHAR	10
id	ID aus Mention	INT	11
beweg	Bewegungen Kennzeichnung	VARCHAR	10
sverk	Sonderverkauf	VARCHAR	10
aellief	für manuellen Preis	VARCHAR	10
alek	manueller Preis	DECIMAL	10,2
aldatum	Datum für manuellen Preis	VARCHAR	30
gewicht	Gewicht	DECIMAL	10,3

Datenbanktabelle "Mastertabelle"**Tabelle B.2:** Datenbanktabelle "Mastertabelle"

Name	Beschreibung	Datentyp	Länge
ArtNr	Artikelnummer	VARCHAR	50
Benennung	Kurzbezeichnung des Artikels	INT	11
Warengruppe	Warengruppennummer	VARCHAR	30
Preis	Endkundenpreis Brutto	VARCHAR	30
EKPreis	Einkaufspreis	BIGINT	13
Hersteller	Nummer des Herstellers	VARCHAR	100
Benennung2	ausführliche Artikelbeschreibung	DECIMAL	15,2
Lieferbkt	Verfügbarkeit des Artikels	TEXT	
Preis2	Händlerpreis Netto	DECIMAL	15,2
hnr	Herstellernummer des Artikels	VARCHAR	30
bild	URL des Artikelbilds	VARCHAR	30
ausbl	ob ein Artikel aus dem Shop ausgeblendet werden soll	INT	11
eol	EOL-Kennzeichnung eines Artikels	VARCHAR	30
fix1	ob der Artikel einen Fixpreis besitzt	VARCHAR	30
sonderpreis	ob der Artikel ein Sonderpreis besitzt	VARCHAR	30
topseller	ob der Artikel ein Topseller ist	VARCHAR	30
itscope	ob der Artikel von IT-Scope verifiziert wurde	INT	11
bez_itscope	ausführliche Artikelbeschreibung von IT-Scope	VARCHAR	10
kurzinfo	Kurzbezeichnung von IT-Scope	VARCHAR	10
datasheet	URL zum Datenblatt des Artikels	INT	11
itscope_bild	URL zum Artikelbild von IT-Scope	VARCHAR	10
qualification	Einstufung der Qualität durch IT-Scope	VARCHAR	10
promo	ob der Artikel ein Promotionartikel des Herstellers ist	VARCHAR	10
neu	ob der Artikel eine Neuheit am Markt ist	DECIMAL	10,2

Datenbanktabelle *TempFavoriten*

Tabelle B.3: Datenbanktabelle *TempFavoriten*

Name	Beschreibung	Datentyp	Länge
Bezeichnung	Kurzbezeichnung des Artikels	VARCHAR	30
Warengruppe	Warengruppennummer	INT	11
hnr	Herstellernummer des Artikels	VARCHAR	50
HerstellerName	Name des Herstellers	VARCHAR	50
HerstellerID	Nummer des Herstellers	INT	11
WarengruppeAnz	Warengruppenname	VARCHAR	50
Nummer	Oberwarengruppennummer	INT	11
OberWG	Oberwarengruppenname	INT	11

C Klassendiagramme

Klassendiagramm @Ausgang

Tabelle C.1: Klassendiagramm @Ausgang

atAusgang
max_last_mod: int hnr_arr: array shop_preis_arr: array shop_preis2_arr: array preis_arr: array brutto_arr: array
atAusgang(\$max_last_mod) add_hnr(\$val) string get_hnr(\$element) add_preis(\$val) string get_preis(\$element) add_brutto(\$val) string get_brutto(\$element) add_shop_preis(\$val) string get_shop_preis(\$element) add_shop_preis2(\$val) string get_shop_preis2(\$element) int get_length() boolean is_in_array(\$hnr) string get_hnr_arr_key(\$hnr) checkpreis(\$preistyp,\$hnr,\$element) int make_timestamp(\$last_mod) int get_max_last_mod()

Klassendiagramm *Favoritenliste*

Tabelle C.2: Klassendiagramm *Favoritenliste*

listeFavoriten
max_last_mod: int hnr_arr: array
listeFavoriten(\$max_last_mod) add_hnr(\$val) string get_hnr(\$element) int get_length() boolean is_in_array(\$hnr) string get_hnr_arr_key(\$hnr) int make_timestamp(\$last_mod) int get_max_last_mod()

Klassendiagramm *Lagerbestand*

Tabelle C.3: Klassendiagramm *Lagerbestand*

bestandLager
max_last_mod: int ArtNr_arr: array bestand_arr: array last_mod_datum_arr: array last_mod_uhrzeit_arr: array Benennung_arr: array
bestandLager(\$max_last_mod) add_ArtNr(\$val) string get_ArtNr(\$element) add_bestand(\$array,\$key,\$val) add_last_mod(\$key,\$val) string get_datum(\$element) string get_uhrzeit(\$element) string get_max_last_mod_datum() string get_max_last_mod_uhrzeit() add_Benennung(\$key,\$val) string get_Benennung(\$element) boolean is_in_array(\$hnr) int get_bestand(\$lager_name,\$ArtNr) int make_timestamp(\$last_mod) int get_max_last_mod()

Klassendiagramm *Onlinebestellung*

Tabelle C.4: Klassendiagramm *Onlinebestellung*

onlineBestellung
\$kundennummer: int \$hnr_arr: array \$stueck_arr: array \$preis_arr_netto: array \$preis_arr_brutto: array \$fix1_arr: array \$sonderpreis_arr: array \$typ: string \$versand_aridnr_arr: array \$nachnahme_aridnr_arr: array \$comment: string \$zeit: string \$datum: string \$anz_pos: int \$bestellid: int
onlineBestellung(\$kundennummer) int get_Kundennummer() add_hnr(\$val) string get_hnr(\$element) add_stueck(\$val) int get_stueck(\$element) add_preisNETTO(\$val) string get_preisNETTO(\$element) add_preisBRUTTO(\$val) string get_preisBRUTTO(\$element) add_fix1(\$val) string get_fix1(\$element) add_sonderpreis(\$val) string get_sonderpreis(\$element) add_typ(\$val) string get_typ() add_comment(\$val) string get_comment() add_zeit(\$val) string get_zeit() add_datum(\$val) string get_datum() add_anzahl_positionen(\$val) string get_anzahl_positionen() add_bestellid(\$val) string get_bestellid() make_netto(\$preis,\$MwSt,\$element) make_brutto(\$preis,\$MwSt,\$element)

D Quelltexte

PHP-Skript für Angebote die mit "@Ausgang" gekennzeichnet sind

```
1 <?php
2
3 //include("classes.php");
4 include("/usr/local/pl-update/bin/Mention2Shop/php/classes.php");
5
6 session_start();
7
8 //include("config.php");
9 include("/usr/local/pl-update/bin/Mention2Shop/php/config.php");
10
11 //wandelt die Variable ttl_Ausgang in Sekunden um für Timestamp vergleich
12 $ttl_atausgang_sek=($ttl_atausgang*24*60*60);
13
14 //aktuelle zeit
15 $timestamp=time();
16
17 //entfernt die Soderzeichen aus dem EMail-Text
18 function MailUml($text){
19     $text=str_replace("&","&",$text);
20     $text=str_replace("&uml;","&#xe4",$text);
21     $text=str_replace("&ouml;","&#xf6",$text);
22     $text=str_replace("&uuml;","&#xfc",$text);
23     $text=str_replace("&Auml;","&#xc4",$text);
24     $text=str_replace("&Ouml;","&#xd6",$text);
25     $text=str_replace("&Uuml;","&#xdc",$text);
26     $text=str_replace("&szlig;","&#xdf",$text);
27     $text=str_replace("&quot;","&quot;",$text);
28     $text=str_replace("&gt;","&gt;",$text);
29     $text=str_replace("&lt;","&lt;",$text);
30     return($text);
31 }
32
33 //initialisiert das SESSION-ARRAY um die Belegnummern zu speichern
34 $_SESSION['bsbelnr_arr']=array();
35
36 //ermittelt alle Blegnummern die @Ausgangmarkiert sind und Zeit und Datum
37 $abfrage_mssql_belege=mssql_query("SELECT distinct (BSBELNR) ,BSKUNDENNR,BSWSZEIT,
    BSDATUM_FROM,$bestkktable WHERE_BSEMENAU=1_AND_BSBELTYP='A'_AND_BSMANKEY
    ='1'_ORDER_BY_BSKUNDENNR,BSBELNR",$dbh1);
38
39 //Laufzahl des Beleges
40 $j=1;
41
42 //solange wie es Angebote gibt die "atAusgang" markiert sind anhand der BSBELNR
    ein Objekt erstellen was alle Daten aufnimmt
43 while($row=mssql_fetch_array($abfrage_mssql_belege)){
44
45     $bsbelnr=$row['BSBELNR'];
46     $kundennr=$row['BSKUNDENNR'];
47     $bswszeit=$row['BSWSZEIT'];
```

```

48 $bsdatum=$row [ 'BSDATUM' ];
49
50 //erstellt ein SESSION-ARRAY um die Belegnummern zu speichern
51 // somit auch nach der WHILE-Schleife zugriff auf die Session-Objekt
52 // anhand der im Array gespeicherten Belegnummer
53 $_SESSION[ 'bsbelnr_arr' ][] = $bsbelnr;
54
55 //## fuer die Ermittlung des TimeStamps aus dem Datum ##
56 $split_bswszeit = preg_split ( "/(..?)/", $bswszeit , -1,
57 PREG_SPLIT_DELIM_CAPTURE|PREG_SPLIT_NO_EMPTY );
58
59 //setzt sekunden immer auf NULL
60 $split_bswszeit [3] = 0;
61
62 $bswszeit_std=$split_bswszeit [0];
63 $bswszeit_min=$split_bswszeit [1];
64 $bswszeit_sek=$split_bswszeit [3];
65
66 $split_bsdatum=explode( ' ', $bsdatum );
67
68 $teile=count( $split_bsdatum );
69
70 //da der SELECT als Monat keine Zahlen sondern 3 Buchstaben liefert
71 // umwandeln der Buchstaben in MonatsZahlen fuer mktime()
72 switch( $split_bsdatum [0] ) {
73     case Jan:
74         $split_bsdatum [0] = 1;
75         break;
76     case Feb:
77         $split_bsdatum [0] = 2;
78         break;
79     case Mar:
80         $split_bsdatum [0] = 3;
81         break;
82     case Apr:
83         $split_bsdatum [0] = 4;
84         break;
85     case May:
86         $split_bsdatum [0] = 5;
87         break;
88     case Jun:
89         $split_bsdatum [0] = 6;
90         break;
91     case Jul:
92         $split_bsdatum [0] = 7;
93         break;
94     case Aug:
95         $split_bsdatum [0] = 8;
96         break;
97     case Sep:
98         $split_bsdatum [0] = 9;
99         break;
100    case Oct:
101        $split_bsdatum [0] = 10;
102        break;
103    case Nov:
104        $split_bsdatum [0] = 11;
105        break;
106    case Dec:
107        $split_bsdatum [0] = 12;
108        break;
109 }
110
111 $bsdatum_jahr=$split_bsdatum [ $teile -2 ];
112 $bsdatum_monat=$split_bsdatum [0];
113 $bsdatum_tag=$split_bsdatum [ $teile -3 ];

```



```

112 //erstellt den UNIX-Timestamp des BelegeDatums (letzte Speicherung)
113 //int mktime ([ int $Stunde [, int $Minute [, int $Sekunde [, int $Monat
114 //            [, int $Tag [, int $Jahr )
115 $bsdatum_last_mod = mktime($bswszeit_std, $bswszeit_min, $bswszeit_sek,
116 $bsdatum_monat, $bsdatum_tag, $bsdatum_jahr);
117
118 //erstellen des atAusgangObjektes mit Name "BSBELNR" und erhält die
119 //Werte (Belegnummer, MentionKundennummer, letztes Belegspeicherdatum)
120 ${"$bsbelnr"} = new favoriten_atausgang($bsbelnr,$kundennr,
121 $bsdatum_last_mod);
122
123 //ermittelt alle Positions-Daten aus den Belegen die mit "atAusgang"
124 //markiert sind
125 $query_positionen=mssql_query("SELECT_BPIDNR, _bpdistrib, _bpbestnr FROM_
126 $bestkptable WHERE_BPBELNR=' '$bsbelnr '" , $dbh1);
127
128 //solange wie Positionen im Angebot vorhanden sind einfüegen der Daten in
129 //das FavoritenObjekt
130 for ($i=0;$i<mssql_num_rows($query_positionen);$i++){
131
132     ${"$bsbelnr"}->add_bpidxnr(mssql_result($query_positionen,$i,0));
133     ${"$bsbelnr"}->add_bpdistrib(mssql_result($query_positionen,$i
134     ,1));
135     ${"$bsbelnr"}->add_bpbestnr(mssql_result($query_positionen,$i,2)
136     );
137
138     //es handelt sich um einen Stamm-Artikel (HNR aus AEL)
139     if(${"$bsbelnr"}->get_bpidxnr($i)!=0){
140
141         $query_hnr_ael = mssql_query("SELECT_ARHERSTNR, _B.
142 BPVKPREIS, _B.BPVKBRUTT FROM_AEL_ as _A, _BESTKP_ as _B, _
143 BESTKK_ as _C WHERE _A.ARIDNR=' ' . ${"$bsbelnr"}->
144 get_bpidxnr($i) . " ' _AND_ _A.ARIDNR=_B.BPIDNR_ AND _C.
145 BSEMENAUS=1_ AND _C.BSBELTYP='A' _AND _C.BSMANKEY='1' _
146 AND _B.BPBELNR=_C.BSBELNR_ AND _B.BPBELNR=' '$bsbelnr '" ,
147 $dbh1);
148
149         //if ((mssql_num_rows($query_hnr_ael)>0) || (mssql_result(
150 $query_hnr_ael,0,0)!="") || (mssql_result(
151 $query_hnr_ael,0,0)!=0)){
152             if(mssql_num_rows($query_hnr_ael)>0){
153                 //hinzufuegen der Herstellernummer in das
154                 //FavoritenObjekt
155                 ${"$bsbelnr"}->add_hnr_ael(${"$bsbelnr"}->
156                 get_bpidxnr($i),mssql_result($query_hnr_ael
157                 ,0,0));
158                 ${"$bsbelnr"}->add_bpvkpreis(${"$bsbelnr"}->
159                 get_bpidxnr($i),mssql_result($query_hnr_ael
160                 ,0,1));
161                 ${"$bsbelnr"}->add_bpvkbrutt(${"$bsbelnr"}->
162                 get_bpidxnr($i),mssql_result($query_hnr_ael
163                 ,0,2));
164             }
165             else{
166                 ${"$bsbelnr"}->add_hnr_ael(${"$bsbelnr"}->
167                 get_bpidxnr($i),"");
168                 echo "KEINE_HNR_in_AEL_ermittelbar_—_
169                 evtl._EOL\n";
170             }
171         }
172     }
173     else{ //es handelt sich um einen Preislisten-Artikel (HNR aus
174         PRL)
175         $query_hnr_prl = mssql_query("SELECT_P.PRHERSTNR, _B.
176 bpvkpreis, _B.bpvkbrutt FROM_PRL_ as _P, _BESTKP_ as _B, _
177 BESTKK_ as _C WHERE _prdistrib=' ' . ${"$bsbelnr"}->
178 get_bpdistrib($i) . " ' _AND_ _prbestnr=' ' . ${"$bsbelnr"}->
179 get_bpbestnr($i) . " ' _AND _B.bpbestnr=_P.prbestnr_ AND _B.

```

```

148         bpdistrib=P.prdistrib_AND_B.BPBELNR=C.BSBELNR_AND_B.
149         BPBELNR='$bsbelnr','$dbh1');
150
151         if(mssql_num_rows($query_hnr_prl)>0){
152             ${"$bsbelnr"}->add_hnr_prl(${"$bsbelnr"}->
153                 get_bpbestnr($i),mssql_result($query_hnr_prl
154                     ,0,0));
155             ${"$bsbelnr"}->add_bpvkpreis(${"$bsbelnr"}->
156                 get_bpbestnr($i),mssql_result($query_hnr_prl
157                     ,0,1));
158             ${"$bsbelnr"}->add_bpvkbrutt(${"$bsbelnr"}->
159                 get_bpbestnr($i),mssql_result($query_hnr_prl
160                     ,0,2));
161         }
162         else{
163             ${"$bsbelnr"}->add_hnr_prl(${"$bsbelnr"}
164                 ->get_bpbestnr($i),"");
165             echo "KEINE_HNR_in_PRL_ermittelbar_—_
166                 evtl._EOL\n";
167         }
168     }
169 }
170
171 //initialisiert das SESSION-ARRAY welches alle AEL bzw. PRL Positionen enthält um
172 //dies per E-Mail zu verschicken das dies KEINE HNR haben
173 $_SESSION['ael_email']=array();
174 $_SESSION['prl_email']=array();
175
176 //solange wie es Belege gibt
177 foreach($_SESSION['bsbelnr_arr'] as $key => $bsbelnr){
178
179     //ermittelt alle in HNR_AEL_ARR enthaltene Positionen ohne HNR und
180     //speichert diese in ein Array (ael_hnr_email[])
181     //und löscht den Schlüssel aus dem ael_hnr_[]
182     ${"$bsbelnr"}->find_ael_hnr_email();
183
184     //ermittelt alle in HNR_PRL_ARR enthaltene Positionen ohne HNR und
185     //speichert diese in ein Array (prl_hnr_email[])
186     //und löscht den Schlüssel aus dem prl_hnr_[]
187     ${"$bsbelnr"}->find_prl_hnr_email();
188
189     if(${"$bsbelnr"}->get_anz_ael_email_arr()>0){
190         foreach(${"$bsbelnr"}->hnr_ael_email_arr as $element => $val){
191
192             //abfrage ob es denn Beleg in dem die HNR steht schon in
193             //DB vorhanden ist
194             $query_beleg_in_db=mysql_query("SELECT_bsbelnr_FROM_
195                 $atausgangstable_WHERE_bsbelnr='$bsbelnr','$dbh2");
196
197             //hinzufügen der HNR in das Array welches alle HNR per
198             //Email versdente wenn Beleg noch nicht existiert
199             //das heist es wird nur dann (1 mal) die HNR versendet
200             //wenn der Beleg neu in die DB angefügt wird
201             if(mysql_num_rows($query_beleg_in_db)==0){
202
203                 //wenn $val noch nicht im Array ist dieses
204                 //anfügen (um nicht 2 mal die selbe ARIDNR
205                 //anzufügen)
206                 if(!in_array("$val", $_SESSION['ael_email'])){
207                     $_SESSION['ael_email'][]=$val;
208                 }
209             }
210         }
211     }
212 }
213
214 if(${"$bsbelnr"}->get_anz_prl_email_arr()>0){

```

```

196         foreach("${bsbelnr"}->hnr_prl_email_arr as $element => $val){
197
198             //abfrage ob es denn Beleg in dem die HNR steht schon in
199             //DB vorhanden ist
200             $query_beleg_in_db=mysql_query("SELECT_bsbelnr_FROM_
201             $atausgangtable_WHERE_bsbelnr='${bsbelnr}', $dbh2);
202
203             //hinzufügen der HNR in das Array welches alle HNR per
204             //Email versendete wenn Beleg noch nicht existiert
205             // das heist es wird nur dann (1 mal) die HNR versendet
206             // wenn der Beleg neu in die DB angefügt wird
207             if(mysql_num_rows($query_beleg_in_db)==0){
208
209                 //wenn $val noch nicht im Array ist dieses
210                 //anfügen (um nicht 2 mal die selbe ARIDNR
211                 //anzufügen)
212                 if(!in_array("$val", $_SESSION['prl_email'])){
213                     $_SESSION['prl_email'][]=$val;
214                 }
215             }
216         }
217     }
218
219     //eintragen aller in AEL enthalten Positionen mit HNR in Nexus-neu.
220     //Kaufhaus.atAusgang
221     if(count("${bsbelnr"}->hnr_ael_arr)>0){
222         foreach("${bsbelnr"}->hnr_ael_arr as $element => $val){
223
224             //Abfrage der MentionFavoriten Tabelle ob die zu
225             //hinzuzufuegenden Datensatze schon vorhanden sind
226             $query_is_in_mention_favoriten = mysql_query("SELECT_
227             bsbelnr_FROM_$atausgangtable_WHERE_bsbelnr='".
228             $bsbelnr."'_AND_bskundennr='".${"$bsbelnr"}->
229             get_kundennummer()."'_AND_hnr='".${"$bsbelnr"}->
230             get_hnr_ael($element)."'", $dbh2);
231
232             //zaehlt die Anzahl der Treffe (ob Datensatz vorhanden
233             //oder nicht)
234             $treffer=mysql_num_rows($query_is_in_mention_favoriten);
235
236             //wenn noch kein Datensatz vorhanden dann einfuegen
237             if($treffer==0){
238
239                 mysql_query("INSERT INTO_$atausgangtable_(
240                     bsbelnr,bskundennr,hnr,bpvkpreis,bpvkbrutt,
241                     last_mod) VALUES('".$bsbelnr."','".$${"$bsbelnr"}->get_kundennummer()."','',${"$bsbelnr"}->get_hnr_ael($element)."','".$${"$bsbelnr"}->get_bpvkpreis($element)."','".$${"$bsbelnr"}->get_bpvkbrutt($element)."','".$${"$bsbelnr"}->get_bsdatum_last_mod()."')",
242                     $dbh2);
243             }
244             else{//wenn Datensatz vorhanden dann Updaten
245
246                 mysql_query("UPDATE_$atausgangtable_SET_bsbelnr
247                     ='".$bsbelnr."'_AND_bskundennr='".${"$bsbelnr"}->get_kundennummer()."'_AND_hnr='".${"$bsbelnr"}->get_hnr_ael($element)."'_AND_bpvkpreis='".${"$bsbelnr"}->get_bpvkpreis($element)."'_AND_bpvkbrutt='".${"$bsbelnr"}->get_bpvkbrutt($element)."'_AND_last_mod='".${"$bsbelnr"}->get_bsdatum_last_mod()."'_WHERE_bsbelnr='".
248                     $bsbelnr."'_AND_bskundennr='".${"$bsbelnr"}->get_kundennummer()."'_AND_hnr='".${"$bsbelnr"}->get_hnr_ael($element)."'", $dbh2);
249             }
250         }
251     }

```

```

231         }
232     }
233 }
234
235 //eintragen aller in PRL enthalten Positionen mit HNR in Nexus-neu.
236 //Kaufhaus.atAusgang
237 if(count("${bsbelnr}"->hnr_prl_arr)>0){
238     foreach("${bsbelnr}"->hnr_prl_arr as $element => $val){
239         //Abfrage der atausgang Tabelle ob die zu hinzufuegenden
240         //Datensaetze schon vorhanden sind
241         $query_is_in_mention_favoriten = mysql_query("SELECT_
242         bsbelnr_FROM_ $atausgangtable_WHERE_ bsbelnr='".
243         $bsbelnr."'_AND_ bskundennr='". "${bsbelnr}"->
244         get_kundennummer()."'_AND_ hnr='". "${bsbelnr}"->
245         get_hnr_prl($element)."'", $dbh2);
246
247         //zaehlt die Anzahl der Treffe (ob Datensatz vorhanden
248         //oder nicht)
249         $treffer=mysql_num_rows($query_is_in_mention_favoriten);
250
251         //wenn noch kein Datensatz vorhanden dann einfuegen
252         if($treffer==0){
253             mysql_query("INSERT INTO_ $atausgangtable_(
254             bsbelnr,bskundennr,hnr,bpvkpreis,bpvkbrutt,
255             last_mod)_VALUES_ (''. $bsbelnr.'',''. "${bsbelnr}"->get_kundennummer()."',''. "${bsbelnr}"->get_hnr_prl($element)."',''. "${bsbelnr}"->get_bpvkpreis($element)."',''. "${bsbelnr}"->get_bpvkbrutt($element)."',''. "${bsbelnr}"->get_bsdatum_last_mod()."')",
256             $dbh2);
257         }
258         else{//wenn Datensatz vorhanden dann Updaten
259             mysql_query("UPDATE_ $atausgangtable_SET_ bsbelnr
260             =''. $bsbelnr.'',_bskundennr='". "${bsbelnr}"->get_kundennummer()."','_hnr='". "${bsbelnr}"->get_hnr_prl($element)."','_bpvkpreis='". "${bsbelnr}"->get_bpvkpreis($element)."','_bpvkbrutt='". "${bsbelnr}"->get_bpvkbrutt($element)."','_last_mod='". "${bsbelnr}"->get_bsdatum_last_mod()."'_WHERE_ bsbelnr='". $bsbelnr.''_AND_ bskundennr='". "${bsbelnr}"->get_kundennummer()."'_AND_ hnr='". "${bsbelnr}"->get_hnr_prl($element)."'", $dbh2);
261         }
262     }
263 }
264
265 //initialisiert das SESSION-ARRAY um die Belegnummern zu speichern
266 $_SESSION['bsbelnr_loesch_arr']=array();
267
268 // ermittelt alle BSBELNR (Belegnummern) die in atausgang sind
269 $query_loesch_belege=mysql_query("SELECT_ distinct (bsbelnr)_FROM_ $atausgangtable"
270 , $dbh2);
271
272 while($row=mysql_fetch_array($query_loesch_belege)){
273     //anfügen alle BSBELNR in ein SESSION-Array
274     $_SESSION['bsbelnr_loesch_arr'][]=$row['bsbelnr'];
275 }
276
277 //solange wie es Elemente im SESSION-Array gibt

```

```

271 foreach($_SESSION['bsbelnr_loesch_arr'] as $key => $bsbelnr){
272
273     //abfrage der Mention-DB ob diese BSBELNR auch noch BSLANGFR='1'
        markeirt ist
274     $query_is_beleg_bslangfr=mssql_query("SELECT_BSKUNDENNR_FROM_
        $bestkktable_WHERE_BSBELNR='$bsbelnr'_AND_BSEMENAU=1",$dbh1);
275
276     //wenn Ergebnis <0 (also Beleg NICHT mehr BSLANGFR='1' markeirt ist)
        alle Positionen mit dieser BSBELNR
277     //aus Kaufhaus.atausgang löschen
278     if(mssql_num_rows($query_is_beleg_bslangfr)<1){
279
280         //löscht alle Daten mit der BSBELNR aus der Kaufhaus.atausgang
281         mysql_query("DELETE_FROM_ $atausgangtable_WHERE_ bsbelnr='$bsbelnr'
            ','',$dbh2);
282     }
283     else{
284         echo "Beleg_". $bsbelnr. "_NICHT_Löschen!_Da_er_noch_BSEMENAU= '1'
            _ist.\n";
285     }
286 }
287
288 //solange wie es Belege gibt
289 foreach($_SESSION['bsbelnr_arr'] as $key => $bsbelnr){
290
291     //errechnung der Endzeit des Beleges
292     $ttl=("${ $bsbelnr}")->get_bsdatum_last_mod()+$ttl_atausgang_sek);
293
294     //ueberprueft ob das aktuelle Datum der Abfrage noch im Zeitschlauch
        BSDATUM + bsp. 60 Tage liegt
295     if($timestamp<=$ttl){
296         echo "Beleg_". $bsbelnr. "_im_Zeitschlauch\n";
297     }
298     else{
299         //löscht alle Daten mit der BSBELNR aus der Kaufhaus.atausgang
300         mysql_query("DELETE_FROM_ $atausgangtable_WHERE_ bsbelnr='$bsbelnr'
            ','',$dbh2);
301     }
302 }
303
304 //solange wie es Belege gibt
305 foreach($_SESSION['bsbelnr_arr'] as $key => $bsbelnr){
306
307     //ermittelt alle HNR aus dem jeweiligen Beleg
308     $query_hrn=mysql_query("SELECT_hnr_FROM_ $atausgangtable_WHERE_ bsbelnr='
        $bsbelnr'", $dbh2);
309
310     //solange wie es HNR pro Beleg gibt
311     while($row = mysql_fetch_array($query_hrn)){
312
313         $hnr=$row['hnr'];
314
315         if(!in_array($hnr,${"$bsbelnr"}->hnr_ael_arr)){
316
317             if(!in_array($hnr,${"$bsbelnr"}->hnr_prl_arr)){
318
319                 //löscht diese HNR aus Kaufhaus.atAusgang
320                 mysql_query("DELETE_FROM_ $atausgangtable_WHERE_ hnr='$hnr'
                    '_AND_ bsbelnr='$bsbelnr'", $dbh2);
321             }
322             else{
323                 echo "HNR_ $hnr_im_PRL_Array\n";
324             }
325         }
326         else{
327             echo "HNR_ $hnr_im_AEL_Array\n";
328         }

```

```

329     }
330 }
331
332 //versenden eineE-Mail an den Betreuer bzw. ProSoft mit den Daten (AEL bpidnr,
      PRL bpdistrib bpbestnr) wo HNR Leer ist
333 if((count($_SESSION['ael_email'])>0)|| (count($_SESSION['prl_email'])>0)){
334
335     $betreff=" Artikel_in_Mention.AEL_bzw_Mention.PRL_ohne_HNR";
336
337     $message="Die_folgenden_Artikel_haben_keine_HNR_im_Mention:\n\n";
338
339     //wenn mindest 1 artikel in EMAIL-ARRAY-AEL ist
340     if(count($_SESSION['ael_email'])>0){
341
342         $message.="in_AEL:\n\n";
343         $i=1;
344
345         foreach($_SESSION['ael_email'] as $key => $val){
346
347             $query_aranummer = mssql_query("SELECT_ARANUMMER_FROM_
              $aeltable_WHERE_ARIDNR='$val'", $dbh1);
348             $message.=" $i )_ARANUMMER: ".mssql_result(
              $query_aranummer,0,0)." \n";
349             $i++;
350         }
351
352         $message.= "\n\n";
353     }
354
355     //wenn mindest 1 artikel in EMAIL-ARRAY-PRL ist
356     if(count($_SESSION['prl_email'])>0){
357
358         $message.="in_PRL:\n\n";
359         $i=1;
360
361         foreach($_SESSION['prl_email'] as $key => $val){
362             $query_prdistrib = mssql_query("SELECT_prdistrib_FROM_
              $prltable_WHERE_prbestnr='$val'", $dbh1);
363             $message.=" $i )_PRBESTNR: $val _PRDISTRIB: ".mssql_result(
              $query_prdistrib,0,0)." \n";
364             $i++;
365         }
366     }
367
368     // Umlaute zurueckverwandeln
369     $message=MailUml($message);
370
371     //versendet die E-Mail
372     mail($admin_mail, $betreff, $message, "FROM: nexus-neu@prosoft-krippner.com
      ");
373 }
374
375 include("clear_hnr.php");
376
377 ?>

```

Quelltext D.1: PHP-Skript für Angebote die mit "@Ausgang" gekennzeichnet sind

PHP-Skript zur Ermittlung der doppelten Herstellernummern für @Ausgang

1 <?php

```

2
3 //initialisiert ein SESSION-Array was alle Kundennummern aus Nexus-neu.Kaufhaus.
  atAusgang beinhaltet
4 $_SESSION['bskundennummer']=array();
5
6 //ermittelt alle BSKUNDENR die atAusgang haben
7 $query_bskundennr=mysql_query("SELECT distinct (bskundennr) FROM $atausgangtable
  ORDER BY bskundennr", $dbh2);
8
9 //solange wie es BSKUNDENNR gibt füllt das SESSION_Array
10 while($row=mysql_fetch_array($query_bskundennr)){
11     $_SESSION['bskundennummer'][]=$row['bskundennr'];
12 }
13
14 foreach($_SESSION['bskundennummer'] as $key => $bskundennr){
15
16     //initialisiert das $hnr_arr für jeden Kunden
17     $hnr_arr=array();
18
19     //ermittelt alle HNR pro BSKUNDENNR die mehr als 1 mal vorhanden sind
20     $query_doppelte_hnr=mysql_query("SELECT hnr, count(hnr) as count_hnr
  FROM $atausgangtable WHERE bskundennr='$bskundennr' GROUP BY hnr
  HAVING count_hnr>1", $dbh2);
21
22     //solange wie es doppelt HNR pro Kunden gibt
23     while($row=mysql_fetch_array($query_doppelte_hnr)){
24
25         $hnr=$row['hnr'];
26
27         $hnr_arr[]=$row['hnr'];
28     }
29
30     foreach($hnr_arr as $key => $hnr){
31
32         //initialisiert das $last_mod für jede HNR
33         $last_mod_arr=array();
34
35         //ermittelt die BSBELN und last_mod von jeder doppelten HNR
36         $query_belege=mysql_query("SELECT bsbelnr, last_mod FROM
  $atausgangtable WHERE hnr='$hnr' AND bskundennr='
  $bskundennr' ORDER BY last_mod", $dbh2);
37
38         while($row=mysql_fetch_array($query_belege)){
39             $bsbelnr=$row['bsbelnr'];
40             $last_mod=$row['last_mod'];
41
42             $last_mod_arr[$bsbelnr]=$last_mod;
43         }
44
45         //entfernt das letzte Element aus dem Array (letztes Element ist
  der jungste Beleg da SELECT nach last_mod sortiert)
46         array_pop($last_mod_arr);
47
48         //entfernt Diese Element aus atAusgang die noch im Array sind da
  dies aeltere Belege sind
49         foreach($last_mod_arr as $bsbelnr => $last_mod){
50
51             mysql_query("DELETE FROM $atausgangtable WHERE last_mod
  ='$last_mod' AND bsbelnr='$bsbelnr' AND hnr='$hnr'",
  $dbh2);
52         }
53     }
54 }
55
56 ?>

```

Quelltext D.2: PHP-Skript zur Ermittlung der doppelten Herstellernummern für @Ausgang

PHP-Skript zum setzen des letzten Änderungsdatums für @Ausgang

```

1 <?php
2
3 //setzt das letzte update Datum
4 mysql_query("UPDATE_ $atausgangtable_SET_last_update='".date("Y-m-d_H:i:s")."',
    $dbh2);
5
6 ?>

```

Quelltext D.3: PHP-Skript zum setzen des letzten Änderungsdatum für @Ausgang

Shell-Skript zur Ermittlung der Änderungsdaten für @Ausgang

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3
4 echo "START: `date`"
5
6 #####atAusgang auf Trinity aktualisieren#####
7
8 #atAusgang-Daten auf Trinity ermitteln (Slave)
9 ssh prosoft@trinity "mysql-h_ $trinityhost -u_ $trinityuser -p$trinitypwd -e '
    SELECT_ bsbelnr , bskundenr , hnr , bpvkpreis , bpvkbrutt , last_mod_FROM_ atAusgang ' _
    $trin
10 ityshopdb" > $homepfad_M2S/atAusgang_trinity.txt
11 sed -i ' ' 's/___/;/g' $homepfad_M2S/atAusgang_trinity.txt
12 sed -i ' ' 'ld' $homepfad_M2S/atAusgang_trinity.txt
13
14 #atAusgang-Daten auf Nexus-Neu ermitteln (Master)
15 $mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'SELECT_ bsbelnr ,
    bskundenr , hnr , bpvkpreis , bpvkbrutt , last_mod_FROM_ atAusgang ' $mysqltestshopdb
16 > $h
17 omepfad_M2S/atAusgang_nexus.txt
18 sed -i ' ' 's/___/;/g' $homepfad_M2S/atAusgang_nexus.txt
19 sed -i ' ' 'ld' $homepfad_M2S/atAusgang_nexus.txt
20
21 sort $homepfad_M2S/atAusgang_trinity.txt > $homepfad_M2S/atAusgang_trinity_sort.
22 txt
23 sort $homepfad_M2S/atAusgang_nexus.txt > $homepfad_M2S/atAusgang_nexus_sort.txt
24 diff $homepfad_M2S/atAusgang_trinity_sort.txt $homepfad_M2S/atAusgang_nexus_sort
25 .txt > $homepfad_M2S/atAusgang_diff.txt
26
27 sed '/<_/!d' $homepfad_M2S/atAusgang_diff.txt > $homepfad_M2S/atAusgang_diff_rem
28 .txt
29 sed '/>_/!d' $homepfad_M2S/atAusgang_diff.txt > $homepfad_M2S/atAusgang_diff_add
30 .txt
31 sed -i ' ' 's/<_//g' $homepfad_M2S/atAusgang_diff_rem.txt
32 sed -i ' ' 's/>_//g' $homepfad_M2S/atAusgang_diff_add.txt
33
34 scp $homepfad_M2S/atAusgang_diff_rem.txt prosoft@trinity:/home/prosoft/tmp/
35 atAusgang_diff_rem.txt
36 scp $homepfad_M2S/atAusgang_diff_add.txt prosoft@trinity:/home/prosoft/tmp/
37 atAusgang_diff_add.txt
38
39 ssh prosoft@trinity "php_ /home/prosoft/bin/atAusgang_update.php"
40
41 #loescht die erstellten Dateien

```



```

35 rm $homepfad_M2S/atAusgang_trinity.txt
36 rm $homepfad_M2S/atAusgang_nexus.txt
37 rm $homepfad_M2S/atAusgang_trinity_sort.txt
38 rm $homepfad_M2S/atAusgang_nexus_sort.txt
39 rm $homepfad_M2S/atAusgang_diff.txt
40 rm $homepfad_M2S/atAusgang_diff_rem.txt
41 rm $homepfad_M2S/atAusgang_diff_add.txt
42
43 echo "ENDE: _ 'date ' "

```

Quelltext D.4: Shell-Skript zur Ermittlung der Änderungsdaten für *@Ausgang*

PHP-Skript zum Import der Änderungsdaten in den Web-Shop für *@Ausgang*

```

1  <?php
2
3  include("/home/prosoft/config/config.php");
4
5  $arr_is_updated=array();
6
7  //einlesen add
8  $row = 1;
9  $handle = fopen("$tmppfad/atAusgang_diff_add.txt", "r");
10
11 while(($data = fgetcsv($handle, 1000, ";")) != FALSE){
12
13     $bsbelnr=$data[0];
14     $bskundennr=$data[1];
15     $hnr=$data[2];
16     $abfrage=mysql_query("SELECT hnr FROM atAusgang WHERE bsbelnr='$bsbelnr'
17         AND bskundennr='$bskundennr' AND hnr='$hnr'", $dbh1);
18
19     if(mysql_num_rows($abfrage)>0){
20
21         echo $row."_-". $hnr."_update\n";
22
23         mysql_query("UPDATE atAusgang SET bpvkpreis='$data[3]',
24             bpvkbrutt='$data[4]', last_mod='$data[5]' WHERE bsbelnr='
25             $bsbelnr' AND bskundennr='$bskundennr' AND hnr='$hnr'", $dbh1
26         );
27         $arr_is_updated[$bsbelnr]=$hnr;
28         $row++;
29     }
30     else{
31         echo $row."_-". $hnr."_insert\n";
32
33         mysql_query("INSERT INTO atAusgang (bsbelnr, bskundennr, hnr,
34             bpvkpreis, bpvkbrutt, last_mod) VALUES('$data[0]', '$data[1]',
35             '$data[2]', '$data[3]', '$data[4]', '$data[5]')", $dbh1);
36
37         $arr_is_updated[$bsbelnr]=$hnr;
38         $row++;
39
40         if(mysql_affected_rows($dbh1)<=0){
41             echo "Fehler insert: _";
42             echo "INSERT INTO atAusgang (bsbelnr, bskundennr, hnr,
43                 bpvkpreis, bpvkbrutt, last_mod) VALUES('$data[0]', '$data
44                 [1]', '$data[2]', '$data[3]', '$data[4]', '$data[5]')\n";
45         }
46     }
47 }

```

```

40
41 print_r($arr_is_updated);
42
43 fclose($handle);
44
45 //einlesen rem
46 $row = 1;
47 $handle = fopen("$tmppfad/atAusgang_diff_rem.txt", "r");
48 while(( $data = fgetcsv($handle, 1000, ";")) != FALSE) {
49     $bsbelnr=$data[0];
50     $bskundennr=$data[1];
51     $hnr=$data[2];
52
53     $abfrage=mysql_query("SELECT distinct (bsbelnr) FROM atAusgang", $dbh1);
54
55     while($row=mysql_fetch_array($abfrage)){
56         $bsbelnr=$row['bsbelnr'];
57         $array=$arr_is_updated[$bsbelnr];
58
59         if(count($array)>0){
60             if(!(in_array($hnr, $array))){
61                 if(mysql_num_rows(mysql_query("SELECT hnr FROM
atAusgang WHERE bsbelnr='$bsbelnr' AND
bskundennr='$bskundennr' AND hnr='$hnr',
$dbh1))>0){
62                     mysql_query("DELETE FROM atAusgang WHERE
bsbelnr='$bsbelnr' AND bskundennr='
bskundennr' AND hnr='$hnr', $dbh1);
63                     echo "DELETE FROM atAusgang WHERE
bsbelnr='$bsbelnr' AND bskundennr='
bskundennr' AND hnr='$hnr'\n";
64                     $row++;
65                 }
66             }
67         }
68         else{
69             if(mysql_num_rows(mysql_query("SELECT hnr FROM atAusgang WHERE
bsbelnr='$bsbelnr' AND bskundennr='$bskundennr' AND hnr='
$hnr', $dbh1))>0){
70                 mysql_query("DELETE FROM atAusgang WHERE bsbelnr
='$bsbelnr' AND bskundennr='$bskundennr' AND
hnr='$hnr', $dbh1);
71                 echo "DELETE FROM atAusgang WHERE bsbelnr='$bsbelnr' AND
bskundennr='$bskundennr' AND hnr='$hnr'\n";
72                 $row++;
73             }
74         }
75     }
76 }
77
78 //setzt das aktuellste Änderungsdatum in last_update für die
Objektaktualisierung im Web-Shop
79 mysql_query("UPDATE atAusgang SET last_update='". date("Y-m-d_H:i:s")."', $dbh1);
80
81 ?>

```

Quelltext D.5: PHP-Skript zum Import der Änderungsdaten in den Web-Shop für *@Ausgang*

Auszug aus der *index.php* des Web-Shops

```

1 //aus performance Gründen erst dann wenn festgestellt wurde das es keine
Hinweise zum anzeigen gibt
2 //bzw. auf "Weiter" geklickt wurde und somit die Hinweise schon angezeigt wurden

```

```

3  if($_SESSION['hat_hinweise']==0){
4
5      //wenn User angemeldet ist
6      if($UserKunde==1){
7          //wenn der angemeldete User berechtigt ist atAusgang-Preise zu sehen
8          if($_SESSION['hat_atAusgang']==1){
9
10             //wenn sich ein User laenger als 10 Minuten im Shop bewegt
11             //wird das Session-Objekt gelöscht um es mit den aktuellsten
12             //Daten wieder zu fuellen
13             if(isset($_SESSION['atAusgang'])) {
14
15                 //ueberprueft die Aktualitaet des Objektes (aufruf
16                 //eine Funktion)
17                 if(CheckAktualitaet(($time_to_akt_at_ausgang*60),
18                     $_SESSION['atAusgang']->get_max_last_mod())){
19                     //löscht das Session-Objekt
20                     unset($_SESSION['atAusgang']);
21                 }
22             }
23
24             //Nur einmal bis Session-Objekt atAusgang angelegt ist (
25             //wird beim abmelden in abmelden.php geleert)
26             if(!isset($_SESSION['atAusgang'])) {
27                 //Abfrage der atAusgangsTabelle nach HNR und Preisen für
28                 //die eingeloggte MentionKundennummer aus
29                 //KundenTabelle anhand des Logins
30                 $query_atAusgang = "SELECT_hnr,_bpvkpreis,_bpvkbrutt_
31                 FROM_$atausgangtable_WHERE_bskundenr='$_SESSION[
32                 MentionKdnr]'";
33
34                 $result_atAusgang = mysql_query($query_atAusgang,$dbh1
35                 );
36
37                 //wenn Abfrage mindestens ein Ergebnis liefert Objekt
38                 //erstellen
39                 if(mysql_num_rows($result_atAusgang)!=0){
40                     //erstellt ein SESSION-Objekt mit allen
41                     //Herstellernummern, Preisen, Bruttopreisen
42                     //die der Eingeloggten $_SESSION[MentionKdnr]
43                     //zugeordnet sind
44                     //mit dem jungsten letzten Aenderungsdatum
45                     $_SESSION['atAusgang'] = new atAusgang(
46                         mysql_result(mysql_query("SELECT_max
47                         (last_update)_FROM_$atausgangtable",
48                         $dbh1),0,0));
49
50                     $i=0;
51
52                     //Füllt das Session-Objekt atAusgang mit Werten
53                     //aus dem SELECT
54                     while($row = mysql_fetch_array($result_atAusgang
55                     )){
56                         //ermittelt den Shop Preis (Preis(Endkunde
57                         //Brutto), Preis2(Händler Netto)) der
58                         //HNR
59                         //zum vergleich ob atAusgangspreis oder
60                         //Shopp Preis angezeigt wird
61                         //wenn Shopp Preis günstiger als
62                         //atAusgangspreis dann Shopp Preis
63                         //anzeigen
64                         //ermittelt alle Daten aus Artikel für die
65                         //hnr die in der tmp-Tabelle stehen um
66                         //sie in der liste anzeigen zu können
67                         $sql_shop_preis = "SELECT_Preis,Preis2_
68                         FROM_$artikeltable_WHERE_hnr='$row[0]'
69                         ";

```

```

43
44         $abfrage=mysql_query($sql_shop_preis,$dbh1
45         );
46         while($row_preis=mysql_fetch_array(
47             $abfrage)){
48             $_SESSION['atAusgang']->
49                 add_shop_preis($row[0],
50                 $row_preis['Preis']);
51             $_SESSION['atAusgang']->
52                 add_shop_preis2($row[0],
53                 $row_preis['Preis2']);
54         }
55         $_SESSION['atAusgang']->add_hnr($row[0]);
56         $_SESSION['atAusgang']->add_preis($row[1]);
57         ;
58         $_SESSION['atAusgang']->add_brutto($row
59         [2]);
60
61         //aufruf der Funktion zum vergleich des
62         //atAusgangspreises mit dem Shoppreis
63         $_SESSION['atAusgang']->checkpreis(
64             $_SESSION['Preis'],$row[0],$i);
65
66         $i++;
67     }
68 }
69 }
70 }
71
72 //wenn User angemeldet ist
73 if($UserKunde==1){
74
75     //wenn der Artikel NICHT aus detail.php hinzugefügt wurde
76     if($tpl_var!="detail"){
77         //Artikel zu den Favoriten hinzufügen Kaufhaus.ShopFavoriten mit
78         //UserID und HNR
79         if($zu_den_favoriten==1){
80
81             if(isset($_SESSION['listeFavoriten'])){
82                 $is_in_array=$_SESSION['listeFavoriten']->
83                     is_in_array($hnr);
84             }
85
86             if($is_in_array!="TRUE"){
87                 //fuegt einen Datensatz hinzu
88                 mysql_query("INSERT INTO $shopfavoritentable_ (
89                     UserID,hnr) VALUES ( '$_SESSION[UserID] ', '
90                     $hnr ')", $dbh1);
91
92                 //echo "INSERT INTO $shopfavoritentable
93                     (UserID,hnr) VALUES ( '$_SESSION[
94                     UserID] ', '$hnr ' )";
95             }
96
97             //wenn ein Objekt existiert
98             if(isset($_SESSION['listeFavoriten'])){
99                 //löscht das Session-Objekt listeFavoriten
100                 unset($_SESSION['listeFavoriten']);
101             }
102         }
103     }
104     else{
105         //abfrage ob der zu hinzufügende Datensatz schon
106         //existiert

```

```

93         if($action==""){
94             if(isset($_SESSION['listeFavoriten'])){
95                 $is_in_array=$_SESSION['listeFavoriten']->
                     is_in_array($hnr);
96             }
97         }
98     }
99     else{
100         if(isset($_SESSION['listeFavoriten'])){
101             $is_in_array=$_SESSION['listeFavoriten']->
                     is_in_array($hnr_zub);
102         }
103     }
104     //wenn der Artikel mit der HNR noch nicht in Favoriten dann
        hinzufügen
105     if($is_in_array!="TRUE"){
106         //fügt einen Datensatz hinzu
107         if($action==""){
108             mysql_query("INSERT INTO _$shopfavoritentable_(
                     UserID , hnr ) VALUES ( '$_SESSION[ UserID] ' , '
                     $hnr ' )" , $dbh1);
109
110             //echo "INSERT INTO $shopfavoritentable
                     (UserID , hnr) VALUES ( '$_SESSION[
                     UserID] ' , '$hnr ' ) ";
111         }
112         elseif($zu_den_favoriten==1){
113             mysql_query("INSERT INTO _$shopfavoritentable_(
                     UserID , hnr ) VALUES ( '$_SESSION[ UserID] ' , '
                     $hnr_zub ' )" , $dbh1);
114
115             //echo "INSERT INTO $shopfavoritentable
                     (UserID , hnr) VALUES ( '$_SESSION[
                     UserID] ' , '$hnr_zub ' ) ";
116         }
117     }
118
119     //wenn ein Objekt existiert
120     if(isset($_SESSION['listeFavoriten'])){
121         //löscht das Session-Objekt listeFavoriten
122         unset($_SESSION['listeFavoriten']);
123     }
124 }
125
126 //zum entferne eines Favoriten aus Kaufhaus.ShopFavoriten
127 if($delete_favorit==1){
128     mysql_query("DELETE FROM _$shopfavoritentable WHERE hnr='$hnr
                     ' AND UserID='$_SESSION[ UserID] ' " , $dbh1);
129
130     //echo "DELETE FROM $shopfavoritentable WHERE hnr='$hnr ' AND
                     UserID='$_SESSION[ UserID] ' ";
131
132     //löscht das Session-Objekt listeFavoriten
133     unset($_SESSION['listeFavoriten']);
134 }
135
136 //wenn sich ein User laenger als 10 Minuten im Shop bewegt
137 //wird das Session-Objekt gelöscht um es mit den aktuellsten Daten
        wieder zu füllen
138 if(isset($_SESSION['listeFavoriten'])){
139
140     //ueberprueft die Aktualitaet des Objektes (aufruf eine
        Funktion)
141     if(CheckAktualitaet(($time_to_akt_liste_favoriten*60) ,
        $_SESSION['listeFavoriten']->get_max_last_mod())){
142         //löscht das Session-Objekt
143         unset($_SESSION['listeFavoriten']);

```

```

144         }
145     }
146
147
148     //Nur einmal bis Session-Objekt listeFavoriten angelegt ist
149     // (wird beim abmelden in abmelden.php geleert und wenn ein
150     // Artikel hinzugefuegt wurde bzw entfernt wurde)
151     if (!isset($_SESSION['listeFavoriten'])) {
152         //wenn User für Favoritenliste aus Mention
153         // freigeschaltet ist
154         if ($_SESSION['hat_Favoritenliste']==1) {
155             //ermittelt alle HNR aus MentionFavoriten &&
156             // ShopFavortien des Users
157             $query_Favoritenliste = "(SELECT_hnr_FROM_
158             $shopfavoritentable_WHERE_UserID='$_SESSION[UserID]
159             | ')_UNION_(SELECT_hnr_FROM_ $mentionfavoritentable_
160             WHERE_bskundennr='$_SESSION[ MentionKdnr ] ' )";
161         }
162         else {
163             //ermittelt alle HNR ShopFavortien des Users
164             $query_Favoritenliste = "SELECT_hnr_FROM_
165             $shopfavoritentable_WHERE_UserID='$_SESSION[UserID] '
166             ";
167         }
168
169         $result_Favoritenliste = mysql_query($query_Favoritenliste);
170
171         //wenn der angemeldete User mindesten 1 Artikel in
172         // Favoritenliste (Shop bzw. Mention) besitzt
173         if (mysql_num_rows($result_Favoritenliste)>0) {
174             //erstellt ein SESSION-Objekt listeFavoriten mit allen
175             // Herstellernummern
176             //die der Eingeloggten User als Favoriten in Shop bzw.
177             // MentionFavoriten eingetragen hat
178             //mit dem jungsten letzten Aenderungsdatum
179             $_SESSION['listeFavoriten'] = new listeFavoriten(
180                 mysql_result(mysql_query("SELECT_max(last_update)_
181                 FROM_ $mentionfavoritentable" , $dbh1), 0, 0));
182
183             //Füllt das Session-Objekt listeFavoriten mit Werten
184             // aus dem SELECT
185             while($row = mysql_fetch_array($result_Favoritenliste)
186             ) {
187                 $_SESSION['listeFavoriten']->add_hnr($row['hnr'
188                 ]);
189             }
190         }
191     }
192
193     //wenn sich ein User laenger als 1 Stunde im Shop bewegt
194     //wird das Session-Objekt gelöscht um es mit den aktuellsten Daten
195     // wieder zu fuellen
196     if (isset($_SESSION['bestandLager'])) {
197         //ueberprueft die Aktualitaet des Objektes (aufruf eine Funktion)
198         if (CheckAktualitaet(($time_to_akt_bestand_lager*60), $_SESSION['
199         bestandLager']->get_max_last_mod())) {
200             //löscht das Session-Objekt
201             unset($_SESSION['bestandLager']);
202         }
203     }
204
205     //Nur einmal bis Session-Objekt bestandLager angelegt ist (wird beim
206     // abmelden in abmelden.php geleert)
207     if (!isset($_SESSION['bestandLager'])) {

```

```

192
193 //setzt anhand des Lager-Arrays den Select zusammen
194 if(count($lager_arr)>0){
195
196     $min_bestand=" (" ;
197
198     $i=0;
199
200     foreach($lager_arr as $key => $lager_name){
201         $suche_nach_lager_name="L.$lager_name, " ;
202
203         if($i==0){
204             $min_bestand=$lager_name.">0";
205         }
206         else{
207             $min_bestand="_OR_". $lager_name.">0";
208         }
209         $i++;
210     }
211     $min_bestand.=") " ;
212 }
213
214 //Abfrage der LagerBestandTabelle nach ArtNr, Beständen und
    Benennungen
215 $query_bestandLager = "SELECT L.last_mod, L.ArtNr, " .
    $suche_nach_lager_name. " A.Benennung, A.bez_itscope, A.itscope
    FROM $lagerbestandtable as L, " $artikeltable as A WHERE
    $min_bestand AND A.ArtNr=L.ArtNr AND L.hnr=A.hnr ORDER BY L.
    last_mod DESC" ;
216
217 //echo "$query_bestandLager<br>";
218
219 $result_bestandLager = mysql_query($query_bestandLager, $dbh1);
220
221 //wenn Abfrage mindestens ein Ergebnis liefert Objekt erstellen
222 if(mysql_num_rows($result_bestandLager)!=0){
223     //erstellt ein SESSION-Objekt mit dem jüngsten letzten
    Aenderungsdatum für alle Artikel ohne Lagerbestand
224     $_SESSION['bestandLager'] = new bestandLager(
        mysql_result(mysql_query("SELECT max(last_mod) FROM
        $lagerbestandtable", $dbh1), 0, 0));
225
226     while($row = mysql_fetch_array($result_bestandLager)){
227         //fügt die ArtNr einem Session-Objekt-Array hinzu
228         $_SESSION['bestandLager']->add_ArtNr($row['ArtNr']);
229         //fügt die letzte Aenderung (last_mod) einem
    Session-Objekt-Array hinzu mit Index-
    Schlüssel ArtNr
230         $_SESSION['bestandLager']->add_last_mod($row['
    ArtNr'], $row['last_mod']);
231
232         $itscope=$row['itscope'];
233
234         //fügt die Benennung des Artikels einem Session-
    Objekt-Array hinzu
235         if($itscope==TRUE){
236             $_SESSION['bestandLager']->add_Benennung($row['
    ArtNr'], $row['bez_itscope']);
237         }
238         else{
239             $_SESSION['bestandLager']->add_Benennung($row['
    ArtNr'], $row['Benennung']);
240         }
241     }
242
243     if(count($lager_arr)>0){

```

```

244         foreach($lager_arr as $key => $lager_name){
245
246             //bewegt den internen Datensatz-Zeiger
                //eines Anfrageergebnisses zum
                //Datensatz mit der übergebenen Nummer
247             //hier wieder auf fetch_array index=0
248             mysql_data_seek($result_bestandLager, 0)
                ;
249
250             while($row = mysql_fetch_array(
                $result_bestandLager)){
251                 //fügt ein Mehrdimensionalen Assoziativen
                //Array jedem Lager die Verfügbarkeit
                //hinzu mit Index-Schlüssel ArtNr
252                 $_SESSION['bestandLager']->
                    add_bestand($lager_name,$row
                    ['ArtNr'],$row[$lager_name])
                    ;
253             }
254         }
255     }
256 }
257 }
258 }

```

Quelltext D.6: Auszug aus der *index.php* des Web-Shops

Funktion zu Ermittlung der Datenaktualität

```

1 function CheckAktualitaet($time_to_akt_in_sec,$last_mod){
2     //erstellt den aktuellen timestamp
3     $akt_time=time();
4
5     //wenn die Aktuelle Zeit minus $time_to_akt_in_sec größer ist als die
        min_last_mod Zeit
6     if(($akt_time-$time_to_akt_in_sec)>$last_mod){
7         return TRUE;
8     }
9 }

```

Quelltext D.7: Funktion zu Ermittlung der Datenaktualität

Shell-Skript mit Aufruf zweier Shell-Skripte für *Favoritenliste*

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3 echo "START: `date`"
4 echo "##### aufruf_build_MentionFavoriten.sh #####"
5 sh $shellpfad_listeFavoriten/build_MentionFavoriten.sh
6 echo "##### aufruf_copy_MentionFavoriten.sh #####"
7 sh $shellpfad_listeFavoriten/copy_MentionFavoriten.sh
8 echo "ENDE: `date`"

```

Quelltext D.8: Shell-Skript mit Aufruf zweier Shell-Skripte für *Favoritenliste*

Shell-Skript mit Aufruf eines PHP-Skripts zur Ermittlung der Daten aus dem Warenwirtschaftssystem für *Favoritenliste*

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3 echo "##### Build_MentionFavoriten_auf_NEXUS-NEU_
   #####"
4 echo "STEP_1_SELECT_BELEGE"
5 $phppfad/php $phppfad_listeFavoriten/select_liste_favoriten.php > $logpfad_php/
   select_liste_favoriten.log
6 echo "STEP_2_CLEAR_HNR"

```

Quelltext D.9: Shell-Skript mit Aufruf eines PHP-Skripts zur Ermittlung der Daten aus dem Warenwirtschaftssystem für *Favoritenliste*

Shell-Skript zur Ermittlung der Änderungsdaten für *Favoritenliste*

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3
4 echo "START: 'date'"
5
6 #####MentionFavoriten auf Trinity aktualisieren#####
7
8 #MentionFavoriten-Daten auf Trinity ermitteln (Slave)
9 ssh prosoft@trinity "mysql-h_$trinityhost -u_$trinityuser -p$trinitypwd -e '
   SELECT_bsbelnr,bskundenr,hnr,last_mod_FROM_MentionFavoriten'$trinityshopdb
   " > $homepfad_M2S/MentionFavoriten_trinity.txt
10 sed -i 's/;;/g' $homepfad_M2S/atAusgang_trinity.txt
11 sed -i 'ld' $homepfad_M2S/atAusgang_trinity.txt
12
13 #MentionFavoriten-Daten auf Nexus-Neu ermitteln (Master)
14 $mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'SELECT_bsbelnr,
   bskundenr,hnr,last_mod_FROM_MentionFavoriten' $mysqltestshopdb >
   $homepfad_M2S/MentionFavoriten_nexus.txt
15 sed -i 's/;;/g' $homepfad_M2S/atAusgang_nexus.txt
16 sed -i 'ld' $homepfad_M2S/atAusgang_nexus.txt
17
18 sort $homepfad_M2S/MentionFavoriten_trinity.txt > $homepfad_M2S/
   MentionFavoriten_trinity_sort.txt
19 sort $homepfad_M2S/MentionFavoriten_nexus.txt > $homepfad_M2S/
   MentionFavoriten_nexus_sort.txt
20 diff $homepfad_M2S/MentionFavoriten_trinity_sort.txt $homepfad_M2S/
   MentionFavoriten_nexus_sort.txt > $homepfad_M2S/MentionFavoriten_diff.txt
21
22 sed '/</!d' $homepfad_M2S/MentionFavoriten_diff.txt > $homepfad_M2S/
   MentionFavoriten_diff_rem.txt
23 sed '/>/!d' $homepfad_M2S/MentionFavoriten_diff.txt > $homepfad_M2S/
   MentionFavoriten_diff_add.txt
24 sed -i 's/</g' $homepfad_M2S/MentionFavoriten_diff_rem.txt
25 sed -i 's/>/g' $homepfad_M2S/MentionFavoriten_diff_add.txt
26
27 scp $homepfad_M2S/MentionFavoriten_diff_rem.txt prosoft@trinity:/home/prosoft/
   tmp/MentionFavoriten_diff_rem.txt
28 scp $homepfad_M2S/MentionFavoriten_diff_add.txt prosoft@trinity:/home/prosoft/
   tmp/MentionFavoriten_diff_add.txt
29

```

```

30 ssh prosoft@trinity "php_/home/prosoft/bin/MentionFavoriten_update.php"
31
32 #loescht die erstellten Dateien
33 #rm $homepfad_ M2S/MentionFavoriten_trinity.txt
34 #rm $homepfad_ M2S/MentionFavoriten_nexus.txt
35 #rm $homepfad_ M2S/MentionFavoriten_trinity_sort.txt
36 #rm $homepfad_ M2S/MentionFavoriten_nexus_sort.txt
37 #rm $homepfad_ M2S/MentionFavoriten_diff.txt
38 #rm $homepfad_ M2S/MentionFavoriten_diff_rem.txt
39 #rm $homepfad_ M2S/MentionFavoriten_diff_add.txt
40
41 echo "ENDE: _ 'date'"

```

Quelltext D.10: Shell-Skript zur Ermittlung der Änderungsdaten für *Favoritenliste*

PHP-Skript für Angebote die mit "langfristiges Angebot" gekennzeichnet sind

```

1 <?php
2
3 //include("classes.php");
4 include("/usr/local/pl-update/bin/Mention2Shop/php/classes.php");
5
6 session_start();
7
8 //include("config.php");
9 include("/usr/local/pl-update/bin/Mention2Shop/php/config.php");
10
11 //wandelt die Variable ttl_listeFavoriten in sekunden um für Timesamp vergleich
12 $ttl_listeFavoriten_sek=($ttl_listeFavoriten*24*60*60);
13
14 //aktuelle zeit
15 $timestamp=time();
16
17 //entfernt die Soderzeichen aus dem EMail-Text
18 function MailUml($text){
19     $text=str_replace("&","&", $text);
20     $text=str_replace("ä","&#xe4", $text);
21     $text=str_replace("ö","&#xf6", $text);
22     $text=str_replace("ü","&#xfc", $text);
23     $text=str_replace("Ä","&#xc4", $text);
24     $text=str_replace("Ö","&#xd6", $text);
25     $text=str_replace("Ü","&#xdc", $text);
26     $text=str_replace("ß","&#xdf", $text);
27     $text=str_replace(""","&quot;", $text);
28     $text=str_replace(">","&gt;", $text);
29     $text=str_replace("<","&lt;", $text);
30     return($text);
31 }
32
33 //initialisiert das SESSION-ARRAY um die Belegnummern zu speichern
34 $_SESSION['bsbelnr_arr']=array();
35
36 //ermittelt alle Belegnummern, Kundennummern die als "Favoriten" mit "
    langfristiges Angebot" markiert sind
37 $query_belege=mssql_query("SELECT distinct (BSBELNR),BSKUNDENNR,BSWSZEIT,BSDATUM,
    FROM_ $bestkktable_WHERE_BSLANGFR='1' _AND_BSBELTYP='A' _AND_BSMANKEY='1' _ORDER
    _BY_BSKUNDENNR,BSBELNR" , $dbh1);
38
39 //solange wie es Angebote gibt die "langfristiges Angebot" markiert sind anhand
    der BSBELNR ein Objekt erstellen was alle Daten aufnimmt
40 while($row=mssql_fetch_array($query_belege)){

```

```

41
42 $bsbelnr=$row[ 'BSBELNR' ];
43 $kundennr=$row[ 'BSKUNDENNR' ];
44 $bswszeit=$row[ 'BSWSZEIT' ];
45 $bsdatum=$row[ 'BSDATUM' ];
46
47 //erstellt ein SESSION-ARRAY um die Belenummern zu speichern
48 // somit auch nach der WHILE-Schleife zugriff auf die Session-Objekt
   anhand der im Array gespeicherten Belegnummer
49 $_SESSION[ 'bsbelnr_arr' ][] = $bsbelnr;
50
51 $split_bswszeit = preg_split ( "/(..?)/", $bswszeit , -1,
   PREG_SPLIT_DELIM_CAPTURE|PREG_SPLIT_NO_EMPTY );
52
53 //setzt sekunden immer auf NULL
54 $split_bswszeit [3] = 0;
55
56 $bswszeit_std=$split_bswszeit [0];
57 $bswszeit_min=$split_bswszeit [1];
58 $bswszeit_sek=$split_bswszeit [3];
59
60 $split_bsdatum=explode( ' ', $bsdatum );
61
62 $teile=count( $split_bsdatum );
63
64 //da der SELECT als Monat keine Zahlen sondern 3 Buchstaben liefert
   umwandeln der Buchstaben in MonatsZahlen fuer mktime()
65 switch( $split_bsdatum [0] ) {
66     case Jan:
67         $split_bsdatum [0] = 1;
68         break;
69     case Feb:
70         $split_bsdatum [0] = 2;
71         break;
72     case Mar:
73         $split_bsdatum [0] = 3;
74         break;
75     case Apr:
76         $split_bsdatum [0] = 4;
77         break;
78     case May:
79         $split_bsdatum [0] = 5;
80         break;
81     case Jun:
82         $split_bsdatum [0] = 6;
83         break;
84     case Jul:
85         $split_bsdatum [0] = 7;
86         break;
87     case Aug:
88         $split_bsdatum [0] = 8;
89         break;
90     case Sep:
91         $split_bsdatum [0] = 9;
92         break;
93     case Oct:
94         $split_bsdatum [0] = 10;
95         break;
96     case Nov:
97         $split_bsdatum [0] = 11;
98         break;
99     case Dec:
100         $split_bsdatum [0] = 12;
101         break;
102 }
103
104 $bsdatum_jahr=$split_bsdatum [ $teile -2 ];

```

```

105 $bsdatum_monat=$split_bsdatum[0];
106 $bsdatum_tag=$split_bsdatum[ $teile -3];
107
108 //erstellt den UNIX-Timestamp des BelegeDatums (letzte Speicherung)
109 //int mktime ([ int $Stunde [, int $Minute [, int $Sekunde [, int
    $Monat [, int $Tag [, int $Jahr )
110 $bsdatum_last_mod = mktime($bswszeit_std, $bswszeit_min, $bswszeit_sek,
    $bsdatum_monat, $bsdatum_tag, $bsdatum_jahr);
111
112 //erstellen des FavoritenObjektes mit Name "BSBELNR" und erhält die
    Werte (Belegnummer, MentionKundennummer, letztes Belegspeicherdatum)
113 ${"$bsbelnr"} = new favoriten_atausgang($bsbelnr,$kundenr,
    $bsdatum_last_mod);
114
115 //ermittelt alle Positions-Daten aus den Belegen die mit "langfristiges
    Angebot" markiert sind
116 $query_positionen=mssql_query("SELECT_BPIDNR,_bpdistrib,_bpbestnr_FROM_
    $bestkptable_WHERE_BPBELNR='${bsbelnr}'",$dbh1);
117
118 //solange wie Positionen im Angbot vorhanden sind einfüegen der Daten in
    das FavoritenObjekt
119 for($i=0;$i<mssql_num_rows($query_positionen);$i++){
120
121     ${"$bsbelnr"}->add_bpidnr(mssql_result($query_positionen,$i,0));
122     ${"$bsbelnr"}->add_bpdistrib(mssql_result($query_positionen,$i
        ,1));
123     ${"$bsbelnr"}->add_bpbestnr(mssql_result($query_positionen,$i,2)
        );
124
125     //es handelt sich um einen Stamm-Artikel (HNR aus AEL)
126     if(${ "$bsbelnr"}->get_bpidnr($i)!=0){
127
128         $query_hnr_ael = mssql_query("SELECT_ARHERSTNR_FROM_
            $aelttable_WHERE_ARIDNR='".${"$bsbelnr"}->get_bpidnr(
            $i)."'",$dbh1);
129
130         if(mssql_num_rows($query_hnr_ael)>0){
131             //hinzufuegen der Herstellernummer in das
                FavoritenObjekt
132             ${"$bsbelnr"}->add_hnr_ael(${ "$bsbelnr"}->
                get_bpidnr($i),mssql_result($query_hnr_ael
                ,0,0));
133         }
134         else{
135             ${"$bsbelnr"}->add_hnr_ael(${ "$bsbelnr"}->
                get_bpidnr($i),0);
136             echo "KEINE_HNR_in_AEL_ermittelbar_---_evtl._EOL\n"
                ;
137         }
138     }
139     else{ //es handelt sich um einen Preislisten-Artikel (HNR aus
        PRL)
140         $query_hnr_prl = mssql_query("SELECT_PRHERSTNR_FROM_
            $prltable_WHERE_prdistrib='".${"$bsbelnr"}->
            get_bpdistrib($i)."'_AND_prbestnr='".${"$bsbelnr"}->
            get_bpbestnr($i)."'",$dbh1);
141
142         if(mssql_num_rows($query_hnr_prl)>0){
143             ${"$bsbelnr"}->add_hnr_prl(${ "$bsbelnr"
                }->get_bpbestnr($i),mssql_result(
                $query_hnr_prl,0,0));
144         }
145         else{
146             ${"$bsbelnr"}->add_hnr_prl(${ "$bsbelnr"
                }->get_bpbestnr($i),"");
147             echo "KEINE_HNR_in_PRL_ermittelbar_---_evtl._EOL\n"
                ;

```

```

148     }
149 }
150 }
151 }
152
153 //initialisiert das SESSION-ARRAY welches alle AEL bzw. PRL Positionen enthält um
dies per E-Mail zu verschicken das dies KEINE HNR haben
154 $_SESSION['ael_email']=array();
155 $_SESSION['prl_email']=array();
156
157 //solange wie es Belege gibt
158 foreach($_SESSION['bsbelnr_arr'] as $key => $bsbelnr){
159
160     //ermittelt alle in HNR AEL ARR enthaltene Positionen ohne HNR und
speichert diese in ein Array (ael_hnr_email[])
161     // und löscht den Schlüssel aus dem ael_hnr []
162     ${"$bsbelnr"}->find_ael_hnr_email();
163
164     //ermittelt alle in HNR PRL ARR enthaltene Positionen ohne HNR und
speichert diese in ein Array (prl_hnr_email[])
165     // und löscht den Schlüssel aus dem prl_hnr []
166     ${"$bsbelnr"}->find_prl_hnr_email();
167
168     if(${"$bsbelnr"}->get_anz_ael_email_arr()>0){
169         foreach(${"$bsbelnr"}->hnr_ael_email_arr as $element => $val){
170
171             //abfrage ob es denn Beleg in dem die HNR steht schon in
DB vorhanden ist
172             $query_beleg_in_db=mysql_query("SELECT_bsbelnr_FROM_
             $mentionfavoritentable_WHERE_bsbelnr='$bsbelnr' ",
             $dbh2);
173
174             //hinzufügen der HNR in das Array welches alle HNR per
Email versdente wenn Beleg noch nicht existiert
175             // das heist es wird nur dann (1 mal) die HNR versendet
wenn der Beleg neu in die DB angefügt wird
176             if(mysql_num_rows($query_beleg_in_db)==0){
177
178                 //wenn $val noch nicht im Array ist dieses
anfügen (um nicht 2 mal die selbe ARIDNR
anzufügen)
179                 if(!in_array("$val", $_SESSION['ael_email'])){
180                     $_SESSION['ael_email'][]=$val;
181                 }
182             }
183         }
184     }
185
186     if(${"$bsbelnr"}->get_anz_prl_email_arr()>0){
187         foreach(${"$bsbelnr"}->hnr_prl_email_arr as $element => $val){
188
189             //abfrage ob es denn Beleg in dem die HNR steht schon in
DB vorhanden ist
190             $query_beleg_in_db=mysql_query("SELECT_bsbelnr_FROM_
             $mentionfavoritentable_WHERE_bsbelnr='$bsbelnr' ",
             $dbh2);
191
192             //hinzufügen der HNR in das Array welches alle HNR per
Email versdente wenn Beleg noch nicht existiert
193             // das heist es wird nur dann (1 mal) die HNR versendet
wenn der Beleg neu in die DB angefügt wird
194             if(mysql_num_rows($query_beleg_in_db)==0){
195
196                 //wenn $val noch nicht im Array ist dieses
anfügen (um nicht 2 mal die selbe ARIDNR
anzufügen)
197                 if(!in_array("$val", $_SESSION['prl_email'])){

```

```

198         $_SESSION[ 'prl_email' ][] = $val;
199     }
200 }
201 }
202 }
203
204 //eintragen aller in AEL enthalten Positionen mit HNR in Nexus-neu.
205 //Kaufhaus.MentionFavoriten
206 if(count(${"$bsbelnr"}->hnr_ael_arr)>0){
207     foreach(${"$bsbelnr"}->hnr_ael_arr as $element => $val){
208
209         //Abfrage der MentionFavoriten Tabelle ob die zu
210         //hinzuzufuegenden Datensaeetze schon vorhanden sind
211         $query_is_in_mention_favoriten = mysql_query("SELECT_
212             bsbelnr FROM_ $mentionfavoritentable WHERE_ bsbelnr='
213             . $bsbelnr . " ' _AND_ bskundennr=' . ${"$bsbelnr"}->
214             get_kundennummer() . " ' _AND_ hnr=' . ${"$bsbelnr"}->
215             get_hnr_ael($element) . " ' ", $dbh2);
216
217         //zaehlt die Anzahl der Treffe (ob Datensatz vorhanden
218         //oder nicht)
219         $treffer=mysql_num_rows($query_is_in_mention_favoriten);
220
221         //wenn noch kein Datensatz vorhanden dann einfuegen
222         if($treffer==0){
223
224             mysql_query("INSERT INTO_ $mentionfavoritentable_
225             (bsbelnr , bskundennr , hnr , last_mod) VALUES_ ( '
226             . $bsbelnr . " ' , ' . ${"$bsbelnr"}->
227             get_kundennummer() . " ' , ' . ${"$bsbelnr"}->
228             get_hnr_ael($element) . " ' , ' . ${"$bsbelnr"}->
229             get_bsdatum_last_mod() . " ' )", $dbh2);
230
231         }
232         else{//wenn Datensatz vorhanden dann Updaten
233
234             mysql_query("UPDATE_ $mentionfavoritentable SET_
235             bsbelnr=' . $bsbelnr . " ' , bskundennr=' . ${"$bsbelnr"}->get_kundennummer() . " ' , hnr=' . ${"$bsbelnr"}->get_hnr_ael($element) . " ' ,
236             last_mod=' . ${"$bsbelnr"}->
237             get_bsdatum_last_mod() . " ' WHERE_ bsbelnr=' .
238             $bsbelnr . " ' _AND_ bskundennr=' . ${"$bsbelnr"}->
239             get_kundennummer() . " ' _AND_ hnr=' . ${"$bsbelnr"}->
240             get_hnr_ael($element) . " ' ", $dbh2);
241
242         }
243     }
244 }
245
246 //eintragen aller in PRL enthalten Positionen mit HNR in Nexus-neu.
247 //Kaufhaus.MentionFavoriten
248 if(count(${"$bsbelnr"}->hnr_prl_arr)>0){
249     foreach(${"$bsbelnr"}->hnr_prl_arr as $element => $val){
250
251         //Abfrage der MentionFavoriten Tabelle ob die zu
252         //hinzufuegenden Datensaeetze schon vorhanden sind
253         $query_is_in_mention_favoriten = mysql_query("SELECT_
254             bsbelnr FROM_ $mentionfavoritentable WHERE_ bsbelnr='
255             . $bsbelnr . " ' _AND_ bskundennr=' . ${"$bsbelnr"}->
256             get_kundennummer() . " ' _AND_ hnr=' . ${"$bsbelnr"}->
257             get_hnr_prl($element) . " ' ", $dbh2);
258
259         //zaehlt die Anzahl der Treffe (ob Datensatz vorhanden
260         //oder nicht)
261         $treffer=mysql_num_rows($query_is_in_mention_favoriten);
262
263         //wenn noch kein Datensatz vorhanden dann einfuegen

```

```

237         if( $treffer==0){
238
239             mysql_query("INSERT INTO $mentionfavoritentable_
                (bsbelnr ,bskundennr ,hnr ,last_mod) VALUES ( ' "
                . $bsbelnr . " ' , ' " . ${" $bsbelnr"}->
                get_kundennummer() . " ' , ' " . ${" $bsbelnr"}->
                get_hnr_prl($element) . " ' , ' " . ${" $bsbelnr"}->
                get_bsdatum_last_mod() . " ' " , $dbh2);
240         }
241         else{//wenn Datensatz vorhanden dann Updaten
242
243             mysql_query("UPDATE $mentionfavoritentable_SET_
                bsbelnr=' " . $bsbelnr . " ' , _bskundennr=' " . ${"
                $bsbelnr"}->get_kundennummer() . " ' , _hnr=' " . ${"
                $bsbelnr"}->get_hnr_prl($element) . " ' , _
                last_mod=' " . ${" $bsbelnr"}->
                get_bsdatum_last_mod() . " ' WHERE bsbelnr=' " .
                $bsbelnr . " ' AND _bskundennr=' " . ${" $bsbelnr"
                }->get_kundennummer() . " ' AND _hnr=' " . ${"
                $bsbelnr"}->get_hnr_prl($element) . " ' " , $dbh2)
                ;
244         }
245     }
246 }
247 }
248
249 //initialisiert das SESSION-ARRAY um die Belegnummern zu speichern
250 $_SESSION['bsbelnr_loesch_arr']=array();
251
252 // ermittelt alle BSBELNR (Belegnummern) die in MentionFavoriten sind
253 $query_loesch_belege=mysql_query("SELECT distinct (bsbelnr) FROM_
    $mentionfavoritentable" , $dbh2);
254
255 while($row=mysql_fetch_array($query_loesch_belege)){
256
257     //anfügen aller BSBELNR in ein SESSION-Array
258     $_SESSION['bsbelnr_loesch_arr'][]=$row['bsbelnr'];
259 }
260
261 //solange wie es Elemente im SESSION-Array gibt
262 foreach($_SESSION['bsbelnr_loesch_arr'] as $key => $bsbelnr){
263
264     //abfrage der Mention-DB ob diese BSBELNR auch noch BSLANGFR='1'
    //markeirt ist
265     $query_is_beleg_bslangfr=mssql_query("SELECT _BSKUNDENNR FROM_
        $bestkktable WHERE _BSBELNR=' $bsbelnr ' AND _BSLANGFR='1' " , $dbh1);
266
267     //wenn Ergebnis <0 (also Beleg NICHT mehr BSLANGFR='1' markeirt ist)
    //alle Positionen mit dieser BSBELNR
268     //aus Kaufhaus.MentionFavoriten löschen
269     if(mssql_num_rows($query_is_beleg_bslangfr)<1){
270         echo "Beleg_ " . $bsbelnr . " _Löschen_ _Da_ er_ nicht_ mehr_ BSLANGFR='1'_
            ist_ \n";
271
272         //löscht alle Daten mit der BSBELNR aus der Kaufhaus.
        //MentionFavoriten
273         mysql_query("DELETE FROM_ $mentionfavoritentable WHERE bsbelnr='
            $bsbelnr ' " , $dbh2);
274     }
275     else{
276         echo "Beleg_ " . $bsbelnr . " _NICHT_ Löschen! _Da_ er_ noch_ BSLANGFR='1'_
            ist_ \n";
277     }
278 }
279
280 //solange wie es Belege gibt
281 foreach($_SESSION['bsbelnr_arr'] as $key => $bsbelnr){

```

```

282
283 //errechnung der Endzeit des Beleges
284 $ttl=${"$bsbelnr"}->get_bsdatum_last_mod()+$ttl_listeFavoriten_sek);
285
286 //ueberprueft ob das aktuelle Datum der Abfrage noch im Zeitschlauch
    BSDATUM + bsp. 60 Tage liegt
287 if($timestamp<=$ttl){
288     echo "Beleg_". $bsbelnr. "_im_Zeitschlauch\n";
289 }
290 else{
291     echo "\n_____
292 .....echo "\nDer Beleg mit der Nummer: ".$bsbelnr." ist aus dem
    Zeitschlauch für MentionFavoriten rausgefallen";
293 .....echo "\n_____
294
295 //löscht alle Daten mit der BSBELNR aus der Kaufhaus.
    MentionFavoriten
296 mysql_query("DELETE FROM $mentionfavoritentable WHERE bsbelnr='
    $bsbelnr '", $dbh2);
297 }
298 }
299 //solange wie es Belege gibt
300 foreach($_SESSION['bsbelnr_arr'] as $key => $bsbelnr){
301
302 //ermittelt alle HNR aus dem jeweiligen Beleg
303 $query_hrn=mysql_query("SELECT_hnr FROM $mentionfavoritentable WHERE
    bsbelnr='$bsbelnr '", $dbh2);
304
305 //solange wie es HNR pro Beleg gibt
306 while($row = mysql_fetch_array($query_hrn)){
307
308     $hnr=$row['hnr'];
309
310     if(!in_array($hnr, ${"$bsbelnr"}->hnr_ael_arr)){
311
312         if(!in_array($hnr, ${"$bsbelnr"}->hnr_prl_arr)){
313
314             echo "HNR_$hnr_weder_im_AEL_noch_im_PRL_Array\n"
315                 ;
316
317             //löscht diese HNR aus Kaufhaus. MentionFavoriten
            mysql_query("DELETE FROM $mentionfavoritentable
                WHERE_hnr='$hnr' _AND_ bsbelnr='$bsbelnr '",
                $dbh2);
318
319             }
320             else{
321                 echo "HNR_$hnr_im_PRL_Array\n";
322             }
323         }
324         else{
325             echo "HNR_$hnr_im_AEL_Array\n";
326         }
327     }
328 }
329 //versenden eineE-Mail an den Betreuer bzw. ProSoft mit den Daten (AEL bpidnr,
    PRL bpdistrib bpbestnr) wo HNR Leer ist
330 if((count($_SESSION['ael_email'])>0)|| (count($_SESSION['prl_email'])>0)){
331
332     $betreff=" Artikel_in_Mention.AEL_bzw_Mention.PRL_ohne_HNR";
333
334     $message="Die_folgenden_Artikel_haben_keine_HNR_im_Mention:\n\n";
335
336 //wenn mindest 1 artikel in EMAIL-ARRAY-AEL ist
337 if(count($_SESSION['ael_email'])>0){
338
339     $message.="in_AEL:_\n\n";

```



```

340         $i=1;
341
342         foreach($_SESSION['ael_email'] as $key => $val){
343
344             $query_aranummer = mssql_query("SELECT_ARANUMMER_FROM_
345                 $aeltable_WHERE_ARIDNR='$val'", $dbh1);
346             $message.=" $i)_ARANUMMER: ".mssql_result(
347                 $query_aranummer,0,0)." \n";
348             $i++;
349         }
350         $message.= "\n\n";
351     }
352     //wenn mindest 1 artikel in EMAIL-ARRAY-PRL ist
353     if(count($_SESSION['prl_email'])>0){
354
355         $message.= "in_PRL: \n\n";
356         $i=1;
357
358         foreach($_SESSION['prl_email'] as $key => $val){
359             $query_prdistrib = mssql_query("SELECT_prdistrib_FROM_
360                 $prltable_WHERE_prbestnr='$val'", $dbh1);
361             $message.=" $i)_PRBESTNR: $val_PRDISTRIB: ".mssql_result(
362                 $query_prdistrib,0,0)." \n";
363             $i++;
364         }
365     }
366     // Umlaute zurueckverwandeln
367     $message=MailUml($message);
368
369     //versendet die E-Mail
370     mail($admin_mail, $betreff, $message, "FROM: _nexus-neu@prosoft-krippner.com
371         ");
372 }
373 include("clear_hnr.php");
374 ?>

```

Quelltext D.11: PHP-Skript für Angebote die mit "langfristiges" Angebot gekennzeichnet sind

PHP-Skript zur Ermittlung der doppelten Herstellernummern für *Favoritenliste*

```

1 <?php
2
3 //initialisiert ein SESSION-Array was alle Kundennummern aus Nexus-neu.Kaufhaus.
4   MentionFavoriten beinhaltet
5 $_SESSION['bskundennummer']=array();
6
7 //ermittelt alle BSKUNDENNR die MentionFavoriten haben
8 $query_bskundennr=mysql_query("SELECT_distinct(bskundennr)_FROM_
9   $mentionfavoritentable_ORDER_BY_bskundennr", $dbh2);
10
11 //solange wie es BSKUNDENNR gibt füllt das SESSION_Array
12 while($row=mysql_fetch_array($query_bskundennr)){
13     $_SESSION['bskundennummer'][]=$row['bskundennr'];
14 }
15

```

```

14 foreach($_SESSION['bskundennummer'] as $key => $bskundennr){
15
16     //initialisiert das $hnr_arr für jeden Kunden
17     $hnr_arr=array();
18
19     //ermittelt alle HNR pro BSKUNDENNR die mehr als 1 mal vorhanden sind
20     $query_doppelte_hnr=mysql_query("SELECT_hnr,_count(hnr)_as_count_hnr_
        FROM_ $mentionfavoritentable_WHERE_hnr_='$bskundennr'_GROUP_BY
        _hnr_HAVING_count_hnr>1",$dbh2);
21
22     //solange wie es doppelt HNR pro Kunden gibt
23     while($row=mysql_fetch_array($query_doppelte_hnr)){
24
25         $hnr=$row['hnr'];
26
27         $hnr_arr[]=$row['hnr'];
28     }
29
30     foreach($hnr_arr as $key => $hnr){
31
32         //initialisiert das $last_mod für jede HNR
33         $last_mod_arr=array();
34
35         //ermittelt die BSBELN und last_mod von jeder doppelten HNR
36         $query_belege=mysql_query("SELECT_bsbelnr,_last_mod_FROM_
        $mentionfavoritentable_WHERE_hnr_='$hnr'_AND_bskundennr=
        '$bskundennr'_ORDER_BY_last_mod",$dbh2);
37
38         while($row=mysql_fetch_array($query_belege)){
39             $bsbelnr=$row['bsbelnr'];
40             $last_mod=$row['last_mod'];
41
42             $last_mod_arr[$bsbelnr]=$last_mod;
43         }
44
45         //entfernt das letzte Element aus dem Array (letztes Element ist
        der jungste Beleg da SELECT nach last_mod sortiert)
46         array_pop($last_mod_arr);
47
48         //entfernt Diese Element aus atAusgang die noch im Array sind da
        dies aeltere Belege sind
49         foreach($last_mod_arr as $bsbelnr => $last_mod){
50
51             mysql_query("DELETE_FROM_ $mentionfavoritentable_WHERE_
        last_mod='$last_mod'_AND_bsbelnr='$bsbelnr'_AND_hnr
        ='$hnr'", $dbh2);
52         }
53     }
54 }
55
56 ?>

```

Quelltext D.12: PHP-Skript zur Ermittlung der doppelten Herstellernummern für *Favoritenliste*

PHP-Skript zum Import der Änderungsdaten in den Web-Shop für *Favoritenliste*

```

1 <?php
2
3 include("/home/prosoft/config/config.php");
4

```

```

5 $arr_is_updated=array();
6
7 //einlesen add
8 $row = 1;
9 $handle = fopen("$tmppfad/MentionFavoriten_diff_add.txt", "r");
10
11 while(($data = fgetcsv($handle, 1000, ";")) != FALSE){
12
13     $bsbelnr=$data[0];
14     $bskundennr=$data[1];
15     $hnr=$data[2];
16     $abfrage=mysql_query("SELECT hnr FROM MentionFavoriten WHERE bsbelnr='
        $bsbelnr' AND bskundennr='$bskundennr' AND hnr='$hnr'", $dbh1);
17
18     if(mysql_num_rows($abfrage)>0){
19
20         echo $row."_-". $hnr."_update\n";
21
22         mysql_query("UPDATE MentionFavoriten SET last_mod='$data[5]'
        WHERE bsbelnr='$bsbelnr' AND bskundennr='$bskundennr' AND
        hnr='$hnr'", $dbh1);
23         $arr_is_updated[$bsbelnr][] = $hnr;
24         $row++;
25     }
26     else{
27         echo $row."_-". $hnr."_insert\n";
28
29         mysql_query("INSERT INTO MentionFavoriten (bsbelnr, bskundennr,
        hnr, last_mod) VALUES('$data[0]', '$data[1]', '$data[2]', '$data
        [3]')", $dbh1);
30
31         $arr_is_updated[$bsbelnr][] = $hnr;
32         $row++;
33
34         if(mysql_affected_rows($dbh1)<=0){
35             echo "Fehler insert: ";
36             echo "INSERT INTO MentionFavoriten (bsbelnr, bskundennr,
        hnr, last_mod) VALUES('$data[0]', '$data[1]', '$data[2]',
        $data[3])\n";
37         }
38     }
39 }
40
41 print_r($arr_is_updated);
42
43 fclose($handle);
44
45 //einlesen rem
46 $row = 1;
47 $handle = fopen("$tmppfad/MentionFavoriten_diff_rem.txt", "r");
48 while(($data = fgetcsv($handle, 1000, ";")) != FALSE) {
49     $bsbelnr=$data[0];
50     $bskundennr=$data[1];
51     $hnr=$data[2];
52
53     $abfrage=mysql_query("SELECT distinct (bsbelnr) FROM MentionFavoriten ,
        $dbh1);
54
55     while($row=mysql_fetch_array($abfrage)){
56         $bsbelnr=$row['bsbelnr'];
57         $array=$arr_is_updated[$bsbelnr];
58
59         if(count($array)>0){
60             if(!(in_array($hnr, $array))){
61                 if(mysql_num_rows(mysql_query("SELECT hnr FROM
        MentionFavoriten WHERE bsbelnr='$bsbelnr' AND bskundennr='$bskundennr' AND
        hnr='$hnr'", $dbh1))>0){

```

```

62 .....mysql_query("DELETE FROM
    MentionFavoriten WHERE bsbelnr='$bsbelnr' AND bskundennr='$bskundennr' AND
    hnr='$hnr'", $dbh1);
63 .....echo "DELETE FROM MentionFavoriten WHERE
    bsbelnr='$bsbelnr' AND bskundennr='$bskundennr' AND hnr='$hnr'\n";
64 .....$row++;
65 .....}
66 .....}
67 .....}
68 .....else {
69 .....if (mysql_num_rows(mysql_query("SELECT hnr FROM
    MentionFavoriten WHERE bsbelnr='$bsbelnr' AND bskundennr='$bskundennr' AND
    hnr='$hnr'", $dbh1))>0){
70 .....mysql_query("DELETE FROM MentionFavoriten WHERE bsbelnr=
    '$bsbelnr' AND bskundennr='$bskundennr' AND hnr='$hnr'", $dbh1);
71 .....echo "DELETE FROM MentionFavoriten WHERE bsbelnr=
    '$bsbelnr' AND bskundennr='$bskundennr' AND hnr='$hnr'\n";
72 .....$row++;
73 .....}
74 .....}
75 .....}
76 }
77
78 //setzt das aktuellste Änderungsdatum in last_update für die
    Objektaktualisierung im Web-Shop
79 mysql_query("UPDATE MentionFavoriten SET last_update='".date("Y-m-d_H:i:s")."',
    $dbh1);
80
81 ?>

```

Quelltext D.13: PHP-Skript zum Import der Änderungsdaten in den Web-Shop für *Favoritenliste*

PHP-Skript zur Auswahl der Ansicht einer für *Favoritenliste*

```

1 <?php
2
3 session_start();
4
5 $tpl->set_block("inhalt", "HatFavoriten", "blockHatFavoriten");
6 $tpl->set_block("inhalt", "HatKeineFavoriten", "blockHatKeineFavoriten");
7
8 //wenn das Session-Objekt existiert
9 if(isset($_SESSION['listeFavoriten'])){
10     //wenn mindestens 1 Artikel als Favorit zur Verfügung steht
11     if(count($_SESSION['listeFavoriten']->hnr_arr)>0){
12
13         $tpl->parse("blockHatFavoriten", "HatFavoriten", true);
14     }
15 }
16 else{
17     $tpl->parse("blockHatKeineFavoriten", "HatKeineFavoriten", true);
18 }
19
20 ?>

```

Quelltext D.14: PHP-Skript zur Auswahl der Ansicht einer für *Favoritenliste*

Shell-Skript zur Ermittlung und Übertragung der aktuellen Lagerbestände

```

1  #!/usr/local/bin/ksh
2  . /usr/local/pl-update/config/config.conf
3
4  echo "START: `date`"
5
6  #####Lagerbestandstabelle auf Nexus-neu aktualisieren#####
7
8  #Ermittlung des Lagerbestand aus Mention
9  $phpfad/php $phpfad_bestandLager/lagerbestand_mention.php > $logpfad_php/
    lagerbestand_mention.log
10
11 #Ermittlung des Lagerbestand aus nexus-neu(nur Stammartikel)
12 $mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'SELECT ArtNr, hnr,
    Delitzsch, Halle, Hauptlager, Lieferbkt, Lieferbkt_Distri FROM LagerBestand_
    WHERE ArtNr>0' $mysqltestshopdb > $homepfad_M2S/
    lagerbestand_nexus_stammartikel.txt
13 sed -i '' 's/\\;/g' $homepfad_M2S/lagerbestand_nexus_stammartikel.txt
14 sed -i '' 'ld' $homepfad_M2S/lagerbestand_nexus_stammartikel.txt
15
16 sort $homepfad_M2S/lagerbestand_nexus_stammartikel.txt > $homepfad_M2S/
    lagerbestand_nexus_stammartikel_sort.txt
17 sort $homepfad_M2S/lagerbestand_mention.txt > $homepfad_M2S/
    lagerbestand_mention_sort.txt
18 diff $homepfad_M2S/lagerbestand_nexus_stammartikel_sort.txt $homepfad_M2S/
    lagerbestand_mention_sort.txt > $homepfad_M2S/lagerbestand_mention_diff.txt
19
20 sed '/</!d' $homepfad_M2S/lagerbestand_mention_diff.txt > $homepfad_M2S/
    lagerbestand_mention_diff_rem.txt
21 sed '/>/!d' $homepfad_M2S/lagerbestand_mention_diff.txt > $homepfad_M2S/
    lagerbestand_mention_diff_add.txt
22 sed -i '' 's/</g' $homepfad_M2S/lagerbestand_mention_diff_rem.txt
23 sed -i '' 's/>/g' $homepfad_M2S/lagerbestand_mention_diff_add.txt
24
25 #Lagerbestand auf nexus aktualisieren
26 $phpfad/php $phpfad_bestandLager/lagerbestand_nexus_update.php
27
28 #####Lagerbestandstabelle auf Trinity aktualisieren#####
29
30 #Ermittlung des Lagerbestand auf Kaufhaus-test (Trinity) Slave
31 ssh prosoft@trinity "mysql -h $trinityhost -u $trinityuser -p$trinitypwd -e '
    SELECT ArtNr, hnr, Delitzsch, Halle, Hauptlager, Lieferbkt FROM LagerBestand_
    $trinityshopdb" > lagerbestand_slave.txt
32 sed -i '' 's/\\;/g' lagerbestand_slave.txt
33 sed -i '' 'ld' lagerbestand_slave.txt
34
35 #Ermittlung des Lagerbestand auf Kaufhaus (Nexus-neu) Master
36 $mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'SELECT ArtNr, hnr,
    Delitzsch, Halle, Hauptlager, Lieferbkt FROM LagerBestand' $mysqltestshopdb >
    lagerbestand_master.txt
37 sed -i '' 's/\\;/g' lagerbestand_master.txt
38 sed -i '' 'ld' lagerbestand_master.txt
39
40 sort lagerbestand_slave.txt > lagerbestand_slave_sort.txt
41 sort lagerbestand_master.txt > lagerbestand_master_sort.txt
42 diff lagerbestand_slave_sort.txt lagerbestand_master_sort.txt >
    lagerbestand_diff.txt
43
44 sed '/</!d' lagerbestand_diff.txt > lagerbestand_diff_rem.txt
45 sed '/>/!d' lagerbestand_diff.txt > lagerbestand_diff_add.txt
46 sed -i '' 's/</g' lagerbestand_diff_rem.txt
47 sed -i '' 's/>/g' lagerbestand_diff_add.txt
48

```

```

49 scp lagerbestand_diff_add.txt prosoft@trinity:/home/prosoft/tmp/
   lagerbestand_diff_add.txt
50 scp lagerbestand_diff_rem.txt prosoft@trinity:/home/prosoft/tmp/
   lagerbestand_diff_rem.txt
51
52 ssh prosoft@trinity "php_/home/prosoft/bin/lagerbestand_update.php"
53
54 ssh prosoft@trinity "mysql_-h_$trinityhost_-u_$trinityuser_-p$trinitypwd_-e_'
   SELECT_ArtNr,hnr,Delitzsch,Halle,Hauptlager,Lieferbkt_FROM_LagerBestand'_
   $trinityshopdb" > lagerbestand_slave_neu.txt
55 sed -i '' 's/;;;/;/g' lagerbestand_slave_neu.txt
56 sed -i '' 'ld' lagerbestand_slave_neu.txt
57 sort lagerbestand_slave_neu.txt > lagerbestand_slave_neu_sort.txt
58
59 diff lagerbestand_slave_neu_sort.txt lagerbestand_master_sort.txt > diff.txt
60
61 rm lagerbestand_diff.txt
62 rm lagerbestand_diff_add.txt
63 rm lagerbestand_diff_rem.txt
64 rm lagerbestand_master.txt
65 rm lagerbestand_master_sort.txt
66 rm lagerbestand_slave.txt
67 rm lagerbestand_slave_sort.txt
68 rm lagerbestand_slave_neu.txt
69 rm lagerbestand_slave_neu_sort.txt
70
71 rm $homepfad_M2S/lagerbestand_mention.txt
72 rm $homepfad_M2S/lagerbestand_mention_diff.txt
73 rm $homepfad_M2S/lagerbestand_mention_diff_add.txt
74 rm $homepfad_M2S/lagerbestand_mention_diff_rem.txt
75 rm $homepfad_M2S/lagerbestand_mention_sort.txt
76 rm $homepfad_M2S/lagerbestand_nexus_stammartikel.txt
77 rm $homepfad_M2S/lagerbestand_nexus_stammartikel_sort.txt
78
79 echo "ENDE: _ `date` "
```

Quelltext D.15: Shell-Skript zur Ermittlung und Übertragung der aktuellen Lagerbestände

PHP-Skript zur Ermittlung der Lagerbestände aus dem Warenwirtschaftssystem

```

1 <?php
2
3 //include("../classes.php");
4 include("/usr/local/pl-update/bin/Mention2Shop/php/classes.php");
5
6 session_start();
7
8 //include("../config.php");
9 include("/usr/local/pl-update/bin/Mention2Shop/php/config.php");
10
11 ### Fuer alle Stammartikel (ArtNr>0) ###
12
13 //ermittelt alle ProSoft-Artikelnummern, hnr aus Nexus.Kauhaus.Artikel
14 $query_ael=mysql_query("SELECT_ArtNr,hnr_FROM_ $artikeltable_where_ArtNr>0_ORDER_
   BY_ArtNr",$dbh2);
15
16 //wenn mindestens 1 Artikel eine ArtNr hat
17 if(mysql_num_rows($query_ael)>0){
18     //initialisiert das Bestands_Array
19     $bestand_arr=array();
20     $hnr_arr=array();
```

```

21
22 //solange wie es Prosoft-Artikel gibt
23 while($row=mysql_fetch_array($query_ael)){
24     $ArtNr=$row['ArtNr'];
25     $hnr=$row['hnr'];
26     $hnr_arr[$hnr]=$ArtNr;
27
28     //abfrage der Lagerbestaende für Stammartiekl aus Mention die mindestens
29     //1 Bestand haben
30     $query_lager_bestand=mssql_query("SELECT_MLLAGER,_MLBESTAND_FROM_
31     $aellagertable_WHERE_MLIDNR=(SELECT_ARIDNR_FROM_$aelttable_WHERE_
32     ARANUMMER='$ArtNr')_AND_MLMANKEY=1_ORDER_BY_MLLAGER",$dbh1);
33
34     //wenn mindestens 1 Lager für den Artikel vorhanden
35     if(mssql_num_rows($query_lager_bestand)>0){
36         while($row2=mssql_fetch_array($query_lager_bestand)){
37             $lager=rtrim($row2['MLLAGER']);
38             $bestand=$row2['MLBESTAND'];
39
40             //wenn mindestens 1 Lager vorhanden
41             if(count($lager_arr)>0){
42                 //für jedes Lager aus dem Array
43                 foreach($lager_arr as $key => $lager_name){
44                     //wenn das Lager aus dem Array gleich dem aus der Zeile
45                     //ist
46                     if(strcmp($lager,$lager_name)==0){
47                         //Lagerbestand dem Lager zugeordnet in ein Bestands-
48                         //Array schreiben
49                         $bestand_arr[$hnr][$lager_name]=$bestand;
50                     }
51                     else{
52                         if($bestand_arr[$hnr][$lager_name]==0){
53                             $bestand_arr[$hnr][$lager_name]=0;
54                         }
55                     }
56                 }
57             }
58         }
59     }
60
61     if(count($bestand_arr)>0){
62         $csv="";
63         $inserts_mit_bestand=0;
64         $inserts_ohne_bestand=0;
65
66         $i=1;
67
68         foreach($bestand_arr as $key => $value){
69             $hat_bestand="FALSE";
70
71             foreach($value as $key2 => $value2){
72                 if(strcmp($key2,"Delitzsch")==0){
73                     $bestand_dz=$value2;
74                 }
75
76                 if(strcmp($key2,"Halle")==0){
77                     $bestand_hal=$value2;
78                 }
79
80                 if(strcmp($key2,"Hauptlager")==0){
81                     $bestand_l=$value2;
82                 }
83
84                 if($value2>0){
85                     $hat_bestand="TRUE";

```

```

83     }
84 }
85
86 if($hat_bestand=="TRUE"){
87     $csv.="".$hnr_arr[$key].";$key;$bestand_dz;$bestand_hal;$bestand_l;
        ab_Lager;".mysql_result(mysql_query("SELECT_Lieferbkt_Distri
        FROM_$lagerbestandtable_WHERE_hnr='$key' _AND_ArtNr=$hnr_arr[$key
        ]",$dbh2),0,0)."\n";
88 }
89 else{
90     $csv.="".$hnr_arr[$key].";$key;0;0;0;".mysql_result(mysql_query("
        SELECT_Lieferbkt_Distri_FROM_$lagerbestandtable_WHERE_hnr='$key'
        _AND_ArtNr=$hnr_arr[$key]",$dbh2),0,0).";".mysql_result(
        mysql_query("SELECT_Lieferbkt_Distri_FROM_$lagerbestandtable_
        WHERE_hnr='$key' _AND_ArtNr=$hnr_arr[$key]",$dbh2),0,0)."\n";
91 }
92 }
93 }
94
95 //schreibt die Variable $csv in eine Datei
96 $datei="$home_M2S/lagerbestand_mention.txt";
97 $d=fopen($datei,"w");
98 fwrite($d,$csv);
99 fclose($d);
100
101 ?>

```

Quelltext D.16: PHP-Skript zur Ermittlung der Lagerbestände aus dem Warenwirtschaftssystem

PHP-Skript zum Import der Änderungsdaten in das Intranet für Lagerbestand

```

1 <?php
2
3 include("/usr/local/pl-update/bin/Mention2Shop/php/config.php");
4
5 $arr_is_updated=array();
6
7 //einlesen add
8 $row = 1;
9 $handle = fopen("$home_M2S/lagerbestand_mention_diff_add.txt", "r");
10
11 while (($data = fgetcsv($handle, 1000, ";")) != FALSE) {
12     $hnr=$data[1];
13     $anr=$data[0];
14
15     $abfrage=mysql_query("SELECT_hnr_FROM_$lagerbestandtable_WHERE_hnr='$hnr' _
        AND_ArtNr=$anr",$dbh2);
16
17     if(mysql_num_rows($abfrage)>0){
18         echo $row."_-_" . $hnr."_update\n";
19         mysql_query("UPDATE_$lagerbestandtable_SET_Delitzsch=$data[2], Halle=
            $data[3], Hauptlager=$data[4], Lieferbkt='$data[5]' _WHERE_hnr='
            $hnr' _AND_ArtNr=$anr",$dbh2);
20         $arr_is_updated[] = $hnr;
21         $row++;
22     }
23     else{
24         echo $row."_-_" . $hnr."_insert\n";
25         mysql_query("INSERT INTO_$lagerbestandtable_(ArtNr,hnr,Delitzsch,
            Halle,Hauptlager,Lieferbkt,Lieferbkt_Distri) VALUES($data[0], '

```



```

        $data[1] ', $data[2], $data[3], $data[4], ' $data[5] ', ' $data[6] ') ",
        $dbh2);
26     $arr_is_updated[] = $hnr;
27     $row++;
28     if(mysql_affected_rows($dbh2) <= 0){
29         echo "Fehler_insert: ";
30         echo "INSERT INTO $lagerbestandtable (ArtNr, hnr, Delitzsch, Halle,
        Hauptlager, Lieferbkt, Lieferbkt_Distri) VALUES ($data[0], '
        $data[1] ', $data[2], $data[3], $data[4], ' $data[5] ', ' $data[6] ') \
        n";
31     }
32 }
33 }
34
35 fclose($handle);
36
37 //einlesen rem
38 $row = 1;
39 $handle = fopen("$home_M2S/lagerbestand_mention_diff_rem.txt", "r");
40
41 while (($data = fgetcsv($handle, 1000, ";")) != FALSE) {
42     $hnr = $data[1];
43     $anr = $data[0];
44
45     if(!(in_array($hnr, $arr_is_updated))){
46         echo $row. " _hnr_loeschen: ". $hnr. " \n";
47         mysql_query("DELETE FROM $lagerbestandtable WHERE hnr=' $hnr ' _AND_ ArtNr=
        $anr", $dbh2);
48         $row++;
49     }
50 }
51
52 fclose($handle);
53
54 ?>

```

Quelltext D.17: PHP-Skript zum Import der Änderungsdaten in das Intranet für Lagerbestand

PHP-Skript zum Import der Änderungsdaten in in den Web-Shop für Lagerbestand

```

1 <?php
2
3 include("/home/prosoft/config/config.php");
4
5 $arr_is_updated = array();
6
7 //einlesen add
8 $row = 1;
9 $handle = fopen("$stmpfad/lagerbestand_diff_add.txt", "r");
10
11 while (($data = fgetcsv($handle, 1000, ";")) != FALSE) {
12     $hnr = $data[1];
13     $anr = $data[0];
14
15     $abfrage = mysql_query("SELECT hnr FROM LagerBestand WHERE hnr=' $hnr ' _AND_
        ArtNr= $anr", $dbh1);
16
17     if(mysql_num_rows($abfrage) > 0){
18         mysql_query("UPDATE LagerBestand SET Delitzsch= $data[2], Halle= $data[3],
        Hauptlager= $data[4], Lieferbkt= ' $data[5] ' WHERE hnr=' $hnr ' _AND_ ArtNr=

```

```

19         $anr", $dbh1);
20         $arr_is_updated[] = $hnr;
21         $row++;
22     }
23     else {
24         mysql_query("INSERT INTO LagerBestand (ArtNr, hnr, Delitzsch, Halle,
25             Hauptlager, Lieferbkt) VALUES($data[0], '$data[1]', $data[2], $data[3],
26             $data[4], '$data[5]')", $dbh1);
27         $arr_is_updated[] = $hnr;
28         $row++;
29
30         if(mysql_affected_rows($dbh1) <= 0) {
31             echo "Fehler insert: ";
32             echo "INSERT INTO LagerBestand (ArtNr, hnr, Delitzsch, Halle, Hauptlager,
33                 Lieferbkt) VALUES($data[0], '$data[1]', $data[2], $data[3], $data
34                 [4], '$data[5]')\n";
35         }
36     }
37 }
38
39 fclose($handle);
40
41 //einlesen rem
42 $row = 1;
43 $handle = fopen("Stmppfad/lagerbestand_diff_rem.txt", "r");
44
45 while (($data = fgetcsv($handle, 1000, ";")) !== FALSE) {
46     $hnr = $data[1];
47     $anr = $data[0];
48     if (!(in_array($hnr, $arr_is_updated))) {
49         echo $row. "hnr löschen: ". $hnr. "\n";
50         mysql_query("DELETE FROM LagerBestand WHERE hnr='$hnr' AND ArtNr=$anr",
51             $dbh1);
52         $row++;
53     }
54 }
55
56 fclose($handle);
57
58 mysql_query("UPDATE LagerBestand SET last_mod='". date("Y-m-d H:i:s") . "' ", $dbh1);
59
60 ?>

```

Quelltext D.18: PHP-Skript zum Import der Änderungsdaten in in den Web-Shop für Lagerbestand

PHP-Funktion zur Ermittlung der Symbole der Verfügbarkeiten

```

1 function Verfuegbarkeit($ArtNr, $Lieferbkt, $tpl_var) {
2
3     global $min_lfz;
4
5     //wenn das SESSION-Objekt existiert
6     if(isset($_SESSION['bestandLager'])) {
7         //wenn die ArtNr in dem Session-Objekt enthalten
8         if($_SESSION['bestandLager']->is_in_array($ArtNr)) {
9             //dann ist eine LagerBestand von mindestens 1 in mindestens 1 Lager
10             $farbe = "gruen";
11             $link_title = "ab Lager ProSoft (bitte Filiale beachten)";
12             $start_text = "ab Lager";
13             $detail_text = "ab Lager ProSoft (bitte Filiale beachten)";

```

```

14     }
15     else{
16         if($Lieferbkt==3){
17             $farbe="gruen_gelb";
18             $link_title="ab_Lager_bei_unserem_Lieferanten_(innerhalb_der_
19                 naechsten_2-3_Werktage_bei_uns)";
20             $start_text="kurzfristig";
21             $detail_text="ab_Lager_Lieferant_(2-3_Werktage)";
22         }
23         elseif($Lieferbkt==2){
24             $farbe="gelb";
25             $link_title="circa_$min_lfz_Werktage_(Angabe_des_Lieferanten)";
26             $start_text="kurzfristig";
27             $detail_text="ca._$min_lfz_Werktage_(Angabe_des_Lieferanten)";
28         }
29         else{
30             $farbe="rot";
31             $link_title="auf_Anfrage";
32             $start_text="auf_Anfrage";
33             $detail_text="auf_Anfrage";
34         }
35     }
36     else{
37         $farbe="rot";
38         $link_title="auf_Anfrage";
39         $start_text="auf_Anfrage";
40         $detail_text="auf_Anfrage";
41     }
42
43     $popup="onClick=\"window.open('','popup', 'width=480,height=260,scrollbars=
44         no,toolbar=no,status=no,resizable=yes,menubar=no,location=no,directories
45         =no,top=300,left=300')\"_href=\"index.php?tpl_var=lager_bestand&ArtNr=
46         $ArtNr\"";
47
48     if($tpl_var!=""){
49         if(($tpl_var=="detail")||($tpl_var=="detail_print")){
50             //für die Detailansicht Große Verfüegbarkeit in der
51             Artikelueberschrift
52             $Verfuegbarkeit_arr[]="<a target=\"popup\"_\".$popup._title=\"\".
53                 $link_title.\"><img id=\"caption\"_src=\"images/\".$farbe.\"
54                 _gross.png\"_width=\"16\"_height=\"16\"_border=\"0\"_alt=\"\"_
55                 style=\"padding-left:4px;padding-bottom:1px;\"/></a>";
56
57             //für die Detailansicht Kleine Verfüegbarkeit in den Artikel-details
58             $Verfuegbarkeit_arr[]="<strong>Verfügbarkeit:</strong><a target=\"
59                 popup\"_\".$popup._title=\"\".$link_title.\"><img class=\"
60                 lbkt_padding\"_src=\"images/\".$farbe.\".png\"_width=\"12\"_height
61                 =\"12\"_border=\"0\"_alt=\"\"_style=\"padding-bottom:2px;_padding
62                 -left:4px;_padding-right:4px;\"_\"/>\".$detail_text.\"</a>";
63
64             //für die Detail-Zubehör-Tabelle
65             $Verfuegbarkeit_arr[]="<a target=\"popup\"_\".$popup._title=\"\".
66                 $link_title.\"><img src=\"images/\".$farbe.\".png\"_width=\"12\"_
67                 height=\"12\"_border=\"0\"_alt=\"\"></a>";
68
69             return $Verfuegbarkeit_arr;
70         }
71         else{
72             //für alle Tabellen (außer Zubehör)
73             return "<a target=\"popup\"_\".$popup._title=\"\".$link_title.\"><
74                 img src=\"images/\".$farbe.\".png\"_width=\"12\"_height=\"12\"_
75                 border=\"0\"_alt=\"\"></a>";
76         }
77     }
78     else{
79         //für die Startseiten-Artikel (Sonderangebote, Topseller)

```

```

65         return "<p style=\"font-size:9px; width:174px; margin-left:2px;\"><
            strong style=\"float:left;\">&nbsp;Verfügbarkeit:</strong><a style
            =\"float:left;\"_target=\"popup\"_\".$popup._title=\"\".$link_title.\"
            \"><img src=\"images/\".$farbe.\".png\"_width=\"12\"_height=\"12\"_
            border=\"0\"_alt=\"\"_style=\"padding-bottom:2px; padding-right:4px;
            _padding-left:4px;\"_\"/><span style=\"position:relative; left:1px;
            text-align:left;\">\".$start_text.\"</span></a></p>";
66     }
67 }

```

Quelltext D.19: PHP-Funktion zur Ermittlung der Symbole der Verfügbarkeiten

PHP-Skript für das Pop-Up-Fenster für *Lagerbestand*

```

1  <?php
2
3  $tpl->set_block("inhalt", "Lager", "blockLager");
4  $tpl->set_block("inhalt", "ProSoftLager", "blockProSoftLager");
5  $tpl->set_block("inhalt", "DistrieLager", "blockDistrieLager");
6
7  //wenn das SESSION-Objekt existiert
8  if(isset($_SESSION['bestandLager'])){
9      //wenn die übergeben ArtNr im Session-Objkt-Array vorhanden ist
10     //also in mindestens 1 Filiale einen Bestand von mindestens 1 hat
11     if($_SESSION['bestandLager']->is_in_array($ArtNr)){
12         $bestandLager=GetLagerBestand($ArtNr);
13         $bestandLagerFiliale=GetLagerBestandFiliale($ArtNr);
14
15         if(count($lager_arr)>0){
16             foreach($lager_arr as $key => $lager_name){
17                 //setzt das Symbol GRÜN
18                 if($bestandLagerFiliale[$lager_name]>0){
19                     $tpl->set_var("symbol", "<img style=\"margin-right:4px;
20                         position:relative; top:2px;\"_src=\"images/gruen.png\"_
21                         width=\"12\"_height=\"12\"_border=\"0\"_alt=\"\">");
22                     $tpl->set_var("verfuegbarkeit", "Verfügbarkeit ab Lager.");
23                 }
24                 else{ //setzt das Symbol GELB
25                     $tpl->set_var("symbol", "<img style=\"margin-right:4px;
26                         position:relative; top:2px;\"_src=\"images/gelb.png\"_
27                         width=\"12\"_height=\"12\"_border=\"0\"_alt=\"\">");
28                     $tpl->set_var("verfuegbarkeit", "Verfügbarkeit kurzfristig
29                         innerhalb 1 Werktag.");
30                 }
31
32                 //schreib "Hauptlager" in "Leipzig" um
33                 if(strcmp($lager_name, "Hauptlager")==0){
34                     $lager_name="Leipzig";
35                 }
36
37                 $tpl->set_var("filiale", $lager_name);
38
39                 $tpl->parse("blockLager", "Lager", true);
40             }
41
42             $tpl->set_var("last_mod_datum", $bestandLager[0]);
43             $tpl->set_var("last_mod_uhrzeit", $bestandLager[1]);
44             $tpl->set_var("artikel", $bestandLager[2]);
45
46             $tpl->parse("blockProSoftLager", "ProSoftLager", true);
47         }
48     }
49     else{

```

```

45     $tpl->set_var("min_lfz", $min_lfz);
46     $tpl->set_var("max_last_mod_datum", $_SESSION['bestandLager']->
        get_max_last_mod_datum());
47     $tpl->set_var("max_last_mod_uhrzeit", $_SESSION['bestandLager']->
        get_max_last_mod_uhrzeit());
48
49     $tpl->parse("blockDistrieLager", "DistrieLager", true);
50 }
51 }
52
53 $tpl->parse("ausgabe", "inhalt");
54
55 ?>

```

Quelltext D.20: PHP-Skript für das Pop-Up-Fenster für *Lagerbestand*

PHP-Funktion zur Ermittlung von Lagebestandsdaten

```

1 function GetLagerBestand($ArtNr){
2     $bestandLager[]=$_SESSION['bestandLager']->get_datum($ArtNr);
3     $bestandLager[]=$_SESSION['bestandLager']->get_uhrzeit($ArtNr);
4     $bestandLager[]=$_SESSION['bestandLager']->get_Benennung($ArtNr);
5
6     return $bestandLager;
7 }

```

Quelltext D.21: PHP-Funktion zur Ermittlung von Lagebestandsdaten

PHP-Funktion zur Ermittlung von Lagerbeständen der Filialen

```

1 function GetLagerBestandFiliale($ArtNr){
2     global $lager_arr;
3
4     if(count($lager_arr)>0){
5         foreach($lager_arr as $key => $lager_name){
6             $bestandLagerFiliale[$lager_name]=$_SESSION['bestandLager']->
                get_bestand($lager_name,$ArtNr);
7         }
8     }
9
10    return $bestandLagerFiliale;
11 }

```

Quelltext D.22: PHP-Funktion zur Ermittlung von Lagerbeständen der Filialen

Template-Datei *lager_bestand.html* für das Pop-Up-Fenster

```

1 <!-- BEGIN ProSoftLager -->
2 <h1>Lagerbestand in den Filialen</h1>
3 <table class="Verfuegbarkeit">
4 <tr>

```

```

5      <th class="text_right">Filiale </td>
6      <th class="text">Verfügbarkeit </td>
7  </tr>
8  <!-- BEGIN Lager -->
9  <tr>
10     <td class="text_left">{filiale}</td>
11     <td class="text_left">{symbol}{verfuegbarkeit}</td>
12 </tr>
13 <!-- END Lager -->
14 </table>
15
16 <p>Letzte Aktualisierung: am {last_mod_datum} um {last_mod_uhrzeit} Uhr</p>
17 <!-- END ProSoftLager -->
18
19 <!-- BEGIN DistrieLager -->
20 <h1>Verfügbarkeiten </h1>
21 <table class="Verfuegbarkeit">
22 <tr>
23   <th class="text_right">Symbol</td>
24   <th class="text">Bedeutung</td>
25 </tr>
26 <tr>
27   <td class="width20"></td>
29   <td class="text_left">Verfügbarkeit: ab Lager ProSoft (bitte Filiale beachten)
30   .</td>
31 </tr>
32 <tr>
33   <td class="width20"></td>
35   <td class="text_left">Verfügbarkeit: ab Lager bei unserem Lieferanten (
36   innerhalb der nächsten 2–3 Werktage bei uns).</td>
37 </tr>
38 <tr>
39   <td class="width20"></td>
41   <td class="text_left">Verfügbarkeit: circa {min_lfz} Werktage (Angabe des
42   Lieferanten).</td>
43 </tr>
44 <tr>
45   <td class="width20"></td>
47   <td class="text_left">Verfügbarkeit: auf Anfrage.</td>
48 </tr>
49 </table>
50
51 <p>Letzte Aktualisierung: am {max_last_mod_datum} um {max_last_mod_uhrzeit} Uhr
52 </p>
53 <!-- END DistrieLager -->

```

Quelltext D.23: Template-Datei *lager_bestand.html* für das Pop-Up-Fenster

Auszug aus dem PHP-Skript *warenkorb.php* zur Erfassung der notwendigen Daten für Onlinebestellung

```

1  .
2  .
3  .
4  //wenn ein SESSION-OBJEKT onlineBestellung existiert (also Kunden Online
5  bestellen kann)
6  if(isset($_SESSION['onlineBestellung'])) {
7      //Abfrage der Positionsdaten aus Tabelle Warenkorb und Artikel

```

```

7  $query="SELECT a.hnr,a.Stueck,b.$_SESSION[Preis]_as_preis,b.fix1,b.
   sonderpreis FROM $warentable_AS_a, $_artikeltable_AS_b WHERE (a.hnr=b.
   hnr) AND (a.UserID=$_SESSION[UserID]) ORDER BY Datum";
8
9  $result = mysql_query($query,$dbh1);
10
11 $_SESSION['onlineBestellung']->add_anzahl_positionen(mysql_num_rows($result)
   );
12
13 $i=0;
14
15 //Füllt das Session-Objekt onlineBestellung mit Werten aus dem SELECT
16 while($row = mysql_fetch_array($result)){
17     //inzufügen der Daten in das Session-Objekt
18     $_SESSION['onlineBestellung']->add_hnr($row[0]);
19     $_SESSION['onlineBestellung']->add_stueck($row[1]);
20     $_SESSION['onlineBestellung']->add_fix1($row[3]);
21     $_SESSION['onlineBestellung']->add_sonderpreis($row[4]);
22
23     #### für die ermittlung des Preises ####
24     //wenn der Kunde für atAusgang freigeschaltet ist dann Preis aus
25     atAusgangObjekt (OHNE evtl. Discount)
26     if(isset($_SESSION['atAusgang'])){
27         $hnr=$_SESSION['onlineBestellung']->get_hnr($i);
28
29         // schaut ob HNR im Objekt vorhanden ist
30         if($_SESSION['atAusgang']->is_in_array($hnr)){
31             //ermittelt Array Key von der HNR
32             $key=$_SESSION['atAusgang']->get_hnr_arr_key($hnr);
33
34             //NettoPreis aus Objekt
35             if($_SESSION['Preis']=="Preis2"){
36                 $_SESSION['onlineBestellung']->add_preisNETTO($_SESSION['
37                     atAusgang']->get_preis($key));
38                 $_SESSION['onlineBestellung']->add_preisBRUTTO($_SESSION['
39                     onlineBestellung']->make_brutto($_SESSION['
40                         onlineBestellung']->get_preisNETTO($i),$MwSt,$i));
41             }
42             else{ //BruttoPreis aus Objekt
43                 $_SESSION['onlineBestellung']->add_preisBRUTTO($_SESSION['
44                     atAusgang']->get_brutto($key));
45                 $_SESSION['onlineBestellung']->add_preisNETTO($_SESSION['
46                     onlineBestellung']->make_netto($_SESSION['
47                         onlineBestellung']->get_preisBRUTTO($i),$MwSt,$i));
48             }
49         }
50         else{
51             if($_SESSION['Preis']=="Preis2"){
52                 //wenn Kein FIX-Preis dann Preis aus Kaufhaus.Artikel mit
53                 evtl. Discount
54                 if((($_SESSION['onlineBestellung']->get_fix1($i)!="TRUE")&&
55                     ($_SESSION['onlineBestellung']->get_sonderpreis($i)!="
56                     TRUE"))){
57                     $_SESSION['onlineBestellung']->add_preisNETTO(
58                         number_format(GetPreis($warengruppe,$row[2]),2,".",
59                         ""));
60                     $_SESSION['onlineBestellung']->add_preisBRUTTO($_SESSION[
61                         'onlineBestellung']->make_brutto($_SESSION['
62                             onlineBestellung']->get_preisNETTO($i),$MwSt,$i));
63                 }
64                 else{ //ansonsten den aus den Preis aus Kaufhaus.Artikel
65                     $_SESSION['onlineBestellung']->add_preisNETTO($row[2]);
66                     $_SESSION['onlineBestellung']->add_preisBRUTTO($_SESSION[
67                         'onlineBestellung']->make_brutto($_SESSION['
68                             onlineBestellung']->get_preisNETTO($i),$MwSt,$i));
69                 }
70             }
71         }
72     }
73 }

```

```

55         else{ // den normalen Preis aus Kaufhaus.Artikel
56             if(($_SESSION['onlineBestellung']->get_fix1($i)!="TRUE")&&
                $_SESSION['onlineBestellung']->get_sonderpreis($i)!="
                TRUE")){
57                 $_SESSION['onlineBestellung']->add_preisBRUTTO(
                    number_format(GetPreis($warengruppe, $row[2]),2,".",
                    ""));
58                 $_SESSION['onlineBestellung']->add_preisNETTO($_SESSION[
                    'onlineBestellung']->make_netto($_SESSION[
                    'onlineBestellung']->get_preisBRUTTO($i),$MwSt,$i));
59             }
60             else{
61                 $_SESSION['onlineBestellung']->add_preisBRUTTO($row[2]);
62                 $_SESSION['onlineBestellung']->add_preisNETTO($_SESSION[
                    'onlineBestellung']->make_netto($_SESSION[
                    'onlineBestellung']->get_preisBRUTTO($i),$MwSt,$i));
63             }
64         }
65     }
66 }
67 else{ // ansonsten den Preis aus der Kaufhaus.Artikel
68     if($_SESSION['Preis']=="Preis2"){
69         //wenn Kein FIX-Preis dann Preis aus Kaufhaus.Artikel mit evtl.
            Discount
70         if(($_SESSION['onlineBestellung']->get_fix1($i)!="TRUE")&&
            $_SESSION['onlineBestellung']->get_sonderpreis($i)!="TRUE"))
71         {
            $_SESSION['onlineBestellung']->add_preisNETTO(number_format(
                GetPreis($warengruppe, $row[2]),2,".", ""));
72         $_SESSION['onlineBestellung']->add_preisBRUTTO($_SESSION[
            'onlineBestellung']->make_brutto($_SESSION[
            'onlineBestellung']->get_preisNETTO($i),$MwSt,$i));
73         }
74         else{ //ansonsten den aus den Preis aus Kaufhaus.Artikel
75             $_SESSION['onlineBestellung']->add_preisNETTO($row[2]);
76             $_SESSION['onlineBestellung']->add_preisBRUTTO($_SESSION[
                'onlineBestellung']->make_brutto($_SESSION[
                'onlineBestellung']->get_preisNETTO($i),$MwSt,$i));
77         }
78     }
79     else{ // den normalen Preis aus Kaufhaus.Artikel
80         if(($_SESSION['onlineBestellung']->get_fix1($i)!="TRUE")&&
            $_SESSION['onlineBestellung']->get_sonderpreis($i)!="TRUE"))
81         {
            $_SESSION['onlineBestellung']->add_preisBRUTTO(number_format(
                GetPreis($warengruppe, $row[2]),2,".", ""));
82         $_SESSION['onlineBestellung']->add_preisNETTO($_SESSION[
            'onlineBestellung']->make_netto($_SESSION[
            'onlineBestellung']->get_preisBRUTTO($i),$MwSt,$i));
83         }
84         else{
85             $_SESSION['onlineBestellung']->add_preisBRUTTO($row[2]);
86             $_SESSION['onlineBestellung']->add_preisNETTO($_SESSION[
                'onlineBestellung']->make_netto($_SESSION[
                'onlineBestellung']->get_preisBRUTTO($i),$MwSt,$i));
87         }
88     }
89 }
90 #####
91 $i++;
92 }
93 }
94 .
95 .
96 .

```


Quelltext D.24: Auszug aus dem PHP-Skript *warenkorb.php* zur Erfassung der notwendigen Daten für Onlinebestellung

PHP-Skript *warenkorb.php* zum Erstellen eines Angebotes oder einer Bestellung für die Onlinebestellung

```

1 <?php
2
3 session_start();
4
5 $tpl->set_block("inhalt", "BestellungSchonGesendetOnline", "
    blockBestellungSchonGesendetOnline");
6 $tpl->set_block("inhalt", "AngebotSchonGesendetOnline", "
    blockAngebotSchonGesendetOnline");
7
8 $tpl->set_block("inhalt", "AnfrageGesendetOnline", "blockAnfrageGesendetOnline")
    ;
9
10 $tpl->set_block("inhalt", "HatNachnahmeOnline", "blockHatNachnahmeOnlineOnline")
    ;
11 $tpl->set_block("inhalt", "HatVersandOnline", "blockHatVersandOnlineOnline");
12 $tpl->set_block("inhalt", "BestellungGesendetContentOnline", "
    blockBestellungGesendetContentOnline");
13 $tpl->set_block("inhalt", "BestellungGesendetOnline", "
    blockBestellungGesendetOnline");
14
15 $tpl->set_block("inhalt", "BestellungNichtGesendetAGBOnline", "
    blockBestellungNichtGesendetAGBOnline");
16 $tpl->set_block("inhalt", "BestellungNichtGesendetOnline", "
    blockBestellungNichtGesendetOnline");
17
18 //erstellt den PreisangabeText je nachdem welche Preisgruppe der angemeldete
    User hat
19 if ($_SESSION['Preis']=="Preis2"){ //User ist Händler
20     $tpl->set_var("preisangabe", "*_Sie_sind_authorisierter_Händler._Alle_
        Artikel-Preise_sind_netto_zuzüglich_MwSt._und_Versandkosten.");
21     $tpl->set_var("preisangabe_zwei_stern", "**_Sie_sind_authorisierter_Händler._
        Alle_Preise_sind_netto_zuzüglich_MwSt..");
22     $tpl->set_var("agb_iframe", "agb_iframe_haend.html");
23 }
24 elseif($_SESSION['Preis']=="EKPreis"){
25     $tpl->set_var("preisangabe", "*_Alle_Artikel-Preise_sind_EK-Preise.");
26     $tpl->set_var("agb_iframe", "agb_iframe_haend.html");
27 }
28 else{
29     $tpl->set_var("preisangabe", "*_Alle_Artikel-Preise_sind_inklusive_19%_MwSt._
        und_zuzüglich_Versandkosten.");
30     $tpl->set_var("preisangabe_zwei_stern", "**_Alle_Preise_sind_inklusive_19%_
        MwSt..");
31     $tpl->set_var("agb_iframe", "agb_iframe_end.html");
32 }
33
34 //wenn ein Session-Objekt gefüllt ist
35 if(isset($_SESSION['onlineBestellung'])){
36     //wenn es eine Anfrage für ein Angebot ist
37     if($Angebot==1){
38         //setzten der des Types (osbeltyp) hier A für Angebot
39         $_SESSION['onlineBestellung']->add_typ('A');
40
41         if(ExistBetreuer($_SESSION['UserID'])){

```

```

42     $tpl->set_var("Filiale", "_an_Ihren_Betreuer_");
43
44     $Fil=GetBetreuerNameReverse($_SESSION['UserID']);
45     $tpl->set_var("Fil", $Fil);
46
47     $Mail=GetBetreuerEmail($_SESSION['UserID']);
48     $tpl->set_var("Mail", $Mail);
49
50     $Betreff="Online-Shop:_Neues_Online-Angebot_in_Mention_eingegangen";
51     $message="Es_ist_ein_Online-Angebot_in_Mention_eingegangen.\n\nBitte
        _prüfen_Sie_dies_im_Mention_unter_Bearbeitung/mention_Online/
        Online_Bestellungen.";
52 }
53
54 $kann_eingetragen_werden=1;
55
56 $tpl->parse("blockAnfrageGesendetOnline", "AnfrageGesendetOnline", true)
57 ;
58 }
59 else{ //es ist eine Bestellung
60     if($agb!="yes"){
61         $tpl->parse("blockBestellungNichtGesendetAGBOnline", "
            BestellungNichtGesendetAGBOnline", true);
62         $tpl->parse("blockBestellungNichtGesendetOnline", "
            BestellungNichtGesendetOnline", true);
63     }
64     else{
65         //setzen der des Types (osbeltyp) hier A für Angebot
66         $_SESSION['onlineBestellung']->add_typ('B');
67
68         if(ExistBetreuer($_SESSION['UserID'])){
69             $tpl->set_var("Filiale", "_an_Ihren_Betreuer_");
70
71             $Fil=GetBetreuerNameReverse($_SESSION['UserID']);
72             $tpl->set_var("Fil", $Fil);
73
74             $Mail=GetBetreuerEmail($_SESSION['UserID']);
75             $tpl->set_var("Mail", $Mail);
76
77             $Betreff="Online-Shop:_Neue_Online-Bestellung_in_Mention_
                eingegangen";
78             $message="Es_ist_eine_Online-Bestellung_in_Mention_eingegangen.\n
                \nBitte_prüfen_Sie_dies_im_Mention_unter_Bearbeitung/
                mention_Online/Online_Bestellungen.";
79         }
80
81         $tpl->set_var("Kundennummer", "<p_class=\"padding\">Ihre_Kundennummer
            _lautet:_<b>$_SESSION[MentionKdnr]</b></p>");
82
83         //anzeige aller Bestellten Positionen im Bestätigungsfenster
84         for($i=0;$i<$_SESSION['onlineBestellung']->get_anzahl_positionen();
            $i++){
85
86             $tpl->set_var("hnr",$_SESSION['onlineBestellung']->get_hnr($i));
87
88             $abfrage=mysql_query("SELECT_Benennung,bez_itscope,itscope_FROM_
                $artikeltable_WHERE_hnr='". $_SESSION['onlineBestellung']->
                get_hnr($i)."'",$dbh1);
89
90             $itscope=mysql_result($abfrage,0,2);
91
92             if($itscope=="TRUE"){
93                 $tpl->set_var("ben",mysql_result($abfrage,0,1));
94             }
95             else{
96                 $tpl->set_var("ben",mysql_result($abfrage,0,0));

```

```

97     }
98
99     if($_SESSION['Preis']=="Preis2"){
100         $tpl->set_var("preis",$_SESSION['onlineBestellung']->
            get_preisNETTO($i));
101     }
102     else{
103         $tpl->set_var("preis",$_SESSION['onlineBestellung']->
            get_preisBRUTTO($i));
104     }
105
106     $tpl->set_var("stck",$_SESSION['onlineBestellung']->get_stueck(
        $i));
107
108     $tpl->parse("blockBestellungGesendetContentOnline", "
        BestellungGesendetContentOnline", true);
109 }
110
111 $tpl->set_var("Bezahlungsart", $zahlung);
112 $tpl->set_var("Versandart", $lieferung);
113
114 // Gesamtpreis ermitteln
115 $gesamtpreis=GesamtPreis($_SESSION['Preis'],$artikeltable,
    $warenktable,$_SESSION['UserID']);
116
117 $gesamtpreis=number_format($gesamtpreis,2,".", "");
118
119 $tpl->set_var("gesamtpreis", $gesamtpreis);
120
121 if($versand!=""){
122     $tpl->set_var("versand", $versand);
123     $tpl->parse("blockHatVersandOnline", "HatVersandOnline", true);
124     $_SESSION['onlineBestellung']->add_versand_aridnr(
        $aridnr_versand);
125
126     if($_SESSION['Preis']=="Preis2"){
127         $_SESSION['onlineBestellung']->add_versand_aridnr($versand_H
            );
128         $_SESSION['onlineBestellung']->add_versand_aridnr(
            make_brutto($versand_H,$MwSt,2));
129     }
130     else{
131         $_SESSION['onlineBestellung']->add_versand_aridnr($versand_E
            );
132         $_SESSION['onlineBestellung']->add_versand_aridnr(
            make_brutto($versand_E,$MwSt,2));
133     }
134     $gesamtbestellpreis_wk=$gesamtpreis+=$versand;
135 }
136
137 if($nachnahme!=""){
138     $tpl->set_var("nachnahme", $nachnahme);
139     $tpl->parse("blockHatNachnahmeOnline", "HatNachnahmeOnline",
        true);
140     $_SESSION['onlineBestellung']->add_nachnahme_aridnr(
        $aridnr_nachnahme);
141
142     if($_SESSION['Preis']=="Preis2"){
143         $_SESSION['onlineBestellung']->add_nachnahme_aridnr(
            $nachnahme_H);
144         $_SESSION['onlineBestellung']->add_nachnahme_aridnr(
            make_brutto($nachnahme_H,$MwSt,2));
145     }
146     else{
147         $_SESSION['onlineBestellung']->add_nachnahme_aridnr(
            $nachnahme_E);

```

```

148         $_SESSION['onlineBestellung']->add_nachnahme_aridnr(
149             make_brutto($nachnahme_E,$MwSt,2));
150     }
151     $gesamtbestellpreis_wk=$gesamtpreis+=$nachnahme;
152 }
153 $gesamtbestellpreis_wk=number_format($gesamtbestellpreis_wk,2,".", "");
154 );
155 $tpl->set_var("gesamtbestellpreis","<p_id=\"gesamtpreis\">Gesamtbestellpreis:&nbsp;
    font-size:16px;\><strong>Gesamtbestellpreis:&nbsp;
    $gesamtbestellpreis_wk&euro;*</strong>&nbsp;&nbsp;<strong>
    style=\"font-size:12px;\></strong></p>");
156
157 //AGB wurden annerkannt und Bestellung kann in Kaufhaus.
    onlineBestellung eingetragen werden
158 $kann_eingetragen_werden=1;
159
160 $tpl->parse("blockBestellungGesendetOnline", "
    BestellungGesendetOnline", true);
161 }
162 }
163
164 //setzen der BestellId (für die Eindeutigkeit (ein Kunde hat mehrere
    Bestellungen))
165 //ermitteln der BestellID aus CRYPT(date("His"),date("Y-m-d"))
166 $_SESSION['onlineBestellung']->add_bestellid(crypt(date("His"),date("Y-m-d")
167 ));
168
169 //setzen des Bermerkung-Textes (oszusatz)
170 $_SESSION['onlineBestellung']->add_comment($Comment);
171
172 //setzen der Bestellzeit (oszeit)
173 $_SESSION['onlineBestellung']->add_zeit(date("His"));
174
175 //setzen des Bestelldatums (osdatum)
176 $_SESSION['onlineBestellung']->add_datum(date("Y-m-d"));
177
178 //versndet eine Info-Mail an den Betreuer das ein Online Angebot/Bestellung
    eingegangen ist
179 Mail($Mail,$Betreff,MailUml($message),"From: ".$Name."<".$eMail.">");
180
181 //um dem Betreuer zum Vergleich der Bestellung die Bestell-EMail über
    sendmail zu zu senden
182 if(file_exists("$php_path/sendmail.php")){
183     include("$php_path/sendmail.php");
184 }
185 }
186 //wenn man F5 drückt wird Sie nicht mehr versendet da nach senden das Session-
    Objekt leer gemacht wurde
187 //nur noch für die Ausschrift
188 else{
189     //wenn es eine Anfrage für ein Angebot ist
190     if($Angebot==1){
191         if ( ExistBetreuer($_SESSION['UserID']) ){
192             $tpl->set_var("Filiale", $_an_Ihren_Betreuer);
193
194             $Fil=GetBetreuerNameReverse($_SESSION['UserID']);
195             $tpl->set_var("Fil", $Fil);
196
197             $Mail=GetBetreuerEmail($_SESSION['UserID']);
198             $tpl->set_var("Mail", $Mail);
199         }
200         $tpl->parse("blockAngebotSchonGesendetOnline", "
            AngebotSchonGesendetOnline", true);
201     }

```

```

202     else{
203         if ( ExistBetreuer($_SESSION['UserID']) ){
204             $tpl->set_var("Filiale", $_an_Ihren_Betreuer_");
205
206             $Fil=GetBetreuerNameReverse($_SESSION['UserID']);
207             $tpl->set_var("Fil", $Fil);
208
209             $Mail=GetBetreuerEmail($_SESSION['UserID']);
210             $tpl->set_var("Mail", $Mail);
211         }
212         $tpl->parse("blockBestellungSchonGesendetOnline", "
                BestellungSchonGesendetOnline", true);
213     }
214 }
215
216 //wenn die AGB bei einer Bestellung anerkannt wurden dann die Bestellung in die
    Kaufhaus.onlineBestellung eintragen
217 if($kann_eingetragen_werden==1){
218     //eintrag der gesammelten Angebotsdaten in die Kaufhaus.onlineBestellung
219     for($i=0;$i<$_SESSION['onlineBestellung']->get_anzahl_positionen();$i++){
220         $bestellid=$_SESSION['onlineBestellung']->get_bestellid();
221         $kundennummer=$_SESSION['onlineBestellung']->get_Kundennummer();
222         $hnr=$_SESSION['onlineBestellung']->get_hnr($i);
223         $typ=$_SESSION['onlineBestellung']->get_typ();
224         $zeit=$_SESSION['onlineBestellung']->get_zeit();
225         $datum=$_SESSION['onlineBestellung']->get_datum();
226         $comment=$_SESSION['onlineBestellung']->get_comment();
227         $preisNETTO=$_SESSION['onlineBestellung']->get_preisNETTO($i);
228         $stueck=$_SESSION['onlineBestellung']->get_stueck($i);
229         $preisBRUTTO=$_SESSION['onlineBestellung']->get_preisBRUTTO($i);
230
231         mysql_query("INSERT INTO _$onlinebestellungtable_(bestellid ,oskundennr ,
                hnr ,osbeltyp ,oszeit ,osdatum ,oszusatz ,opvkpreis ,opstueck ,opvkbrutt ,
                Name)_VALUES_ ( '$bestellid ' , '$kundennummer ' , '$hnr ' , '$typ ' , '$zeit ' ,
                '$datum ' , '$comment ' , '$preisNETTO ' , '$stueck ' , '$preisBRUTTO ' , '$_SESSION
                [name] ' ) " , $dbh1);
232     }
233
234     //eintragen der VersandPosition
235     if($versand!=""){
236         $bestellid=$_SESSION['onlineBestellung']->get_bestellid();
237         $kundennummer=$_SESSION['onlineBestellung']->get_Kundennummer();
238         $aridnr=$_SESSION['onlineBestellung']->get_versand_aridnr(0);
239         $typ=$_SESSION['onlineBestellung']->get_typ();
240         $zeit=$_SESSION['onlineBestellung']->get_zeit();
241         $datum=$_SESSION['onlineBestellung']->get_datum();
242         $comment=$_SESSION['onlineBestellung']->get_comment();
243         $stueck=1;
244         $preisNETTO=$_SESSION['onlineBestellung']->get_versand_aridnr(1);
245         $preisBRUTTO=$_SESSION['onlineBestellung']->get_versand_aridnr(2);
246
247         mysql_query("INSERT INTO _$onlinebestellungtable_(bestellid ,oskundennr ,
                aridnr ,osbeltyp ,oszeit ,osdatum ,oszusatz ,opvkpreis ,opstueck ,
                opvkbrutt ,Name)_VALUES_ ( '$bestellid ' , '$kundennummer ' , '$aridnr ' , '$typ
                ' , '$zeit ' , '$datum ' , '$comment ' , '$preisNETTO ' , '$stueck ' , '$preisBRUTTO
                ' , '$_SESSION[name] ' ) " , $dbh1);
248     }
249
250     //eintragen der NachnahmePosition
251     if($nachnahme!=""){
252         $bestellid=$_SESSION['onlineBestellung']->get_bestellid();
253         $kundennummer=$_SESSION['onlineBestellung']->get_Kundennummer();
254         $aridnr=$_SESSION['onlineBestellung']->get_nachnahme_aridnr(0);
255         $typ=$_SESSION['onlineBestellung']->get_typ();
256         $zeit=$_SESSION['onlineBestellung']->get_zeit();
257         $datum=$_SESSION['onlineBestellung']->get_datum();
258         $comment=$_SESSION['onlineBestellung']->get_comment();

```

```

259     $stueck=1;
260     $preisNETTO=$ _SESSION[ 'onlineBestellung ']->get_nachnahme_aridnr(1);
261     $preisBRUTTO=$ _SESSION[ 'onlineBestellung ']->get_nachnahme_aridnr(2);
262
263     mysql_query("INSERT INTO $onlinebestellungtable (bestellid ,oskundenr ,
        aridnr ,osbeltyp ,oszeit ,osdatum ,oszusatz ,opvkpreis ,opstueck ,
        opvkbrutt ,Name) VALUES ( '$bestellid ' , '$kundennummer ' , '$aridnr ' , '$typ
        ' , '$zeit ' , '$datum ' , '$comment ' , '$preisNETTO ' , '$stueck ' , '$preisBRUTTO
        ' , '$ _SESSION[name] ' ) " , $dbh1);
264 }
265
266 //leert das Session-Objekt nach dem versenden um wenn mann F5 drückt dies
    nicht nochmal macht
267 unset ($ _SESSION[ 'onlineBestellung ']);
268 }
269
270 //Löschen des Warenkorbes nach Bestellung wenn das Häkschen gesetzt wurde
271 if ($wk_loeschen=="yes"){
272     mysql_query("DELETE FROM $warenktable WHERE UserID='$ _SESSION[ UserID ] ' " ,
        $dbh1);
273
274 //löscht das Session-Objekt onlineBestellung
275 unset ($ _SESSION[ 'onlineBestellung ']);
276 }
277
278 ?>

```

Quelltext D.25: PHP-Skript *warenkorb.php* zum Erstellen eines Angebotes oder einer Bestellung für die Onlinebestellung

Shell-Skript zur Übertragung und Import der Onlinebestellungen

```

1  #!/usr/local/bin/ksh
2  . /usr/local/pl-update/config/config.conf
3  echo "START: `date`"
4  echo "###_aufruf_copy_OnlineBestellung.sh_###"
5  sh $shellpfad_importBestellung/copy_OnlineBestellung.sh
6  echo "###_aufruf_import_OnlineBestellung.sh_###"
7  sh $shellpfad_importBestellung/import_OnlineBestellung.sh
8  echo "ENDE: `date`"

```

Quelltext D.26: Shell-Skript zur Übertragung und Import der Onlinebestellungen

Shell-Skript zur Übertragung der Daten für die Onlinebestellungen

```

1  #!/usr/local/bin/ksh
2  . /usr/local/pl-update/config/config.conf
3  echo "###_kopiert die TRINITY.$trinityshopdb.OnlineBestellung_auf_NEXUS-NEU_###"
4  echo ""
5  echo "###_SSH-Verbindung_zu_TRINITY_herstellen_###"
6  echo "###_SELECT_*_FROM_OnlineBestellung_auf_TRINITY_###"
7  ssh prosoft@trinity "mysql-h $trinityhost -u $trinityuser -p$trinitypwd -e '
    SELECT_*_FROM_OnlineBestellung '$trinityshopdb'>$tmppfad/OnlineBestellung.
    txt"
8  echo "###_Leeren der Tabelle_OnlineBestellung_auf_TRINITY_###"

```

```

9 ssh_prosoft@trinity "mysql -h $trinityhost -u $trinityuser -p$trinitypwd -e '
  TRUNCATE OnlineBestellung' $trinityshopdb"
10 echo "### entfernen der ersten Zeile in OnlineBestellung.txt auf TRINITY ###"
11 ssh_prosoft@trinity "sed -i '1d' $tmppfad/OnlineBestellung.txt"
12 echo "### entfernen der Tabulatoren und ersetzten mit ;# in OnlineBestellung.
  txt auf TRINITY ###"
13 ssh_prosoft@trinity "sed -i 's/;/#/g' $tmppfad/OnlineBestellung.txt"
14 echo "### einfüegen von # am Anfang einer jeden Zeile in OnlineBestellung.txt auf
  TRINITY ###"
15 ssh_prosoft@trinity "sed -i 's/^/#/g' $tmppfad/OnlineBestellung.txt"
16 echo "### einfüegen von # an das Ende einer jeden Zeile in OnlineBestellung.txt auf
  TRINITY ###"
17 ssh_prosoft@trinity "sed -i 's/$/#/g' $tmppfad/OnlineBestellung.txt"
18 echo "### kopieren der OnlineBestellung.txt von TRINITY auf Nexus-Neu ###"
19 ssh_prosoft@trinity "sed '$_' $tmppfad/OnlineBestellung.txt" > $homepfad_M2S/
  OnlineBestellung.txt
20 echo "### Kaufhaus. OnlineBestellung auf Nexus-Neu leeren ###"
21 $mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'TRUNCATE
  OnlineBestellung' $mysqltestshopdb
22 echo "### OnlineBestellung.txt in Nexus-Neu. $mysqltestshopdb importieren ###"
23 $mysqlpfad/mysqlimport --local -h $mysqlhost -u $mysqluser -p$mysqlpwd --
  fields-terminated-by=';' --lines-terminated-by='\n' --columns bestellid ,
  oskundennr , hnr , aridnr , osbeltyp , oszeit , osdatum , oszusatz , opvkpreis , opstueck ,
  opvkbrutt , Name $mysqltestshopdb $homepfad_M2S/OnlineBestellung.txt
24 $mysqlpfad/mysqlimport --local -h $mysqlhost -u $mysqluser -p$mysqlpwd --fields-
  enclosed-by='#' --fields-terminated-by=';' --lines-terminated-by='\n' --
  columns bestellid , oskundennr , hnr , aridnr , osbeltyp , oszeit , osdatum , oszusatz ,
  opvkpreis , opstueck , opvkbrutt , Name $mysqltestshopdb $homepfad_M2S/
  OnlineBestellung.txt

```

Quelltext D.27: Shell-Skript zur Übertragung der Daten für die Onlinebestellungen

Shell-Skript zum Import der Daten für die Onlinebestellungen

```

1 #!/usr/local/bin/ksh
2 . /usr/local/pl-update/config/config.conf
3 echo "### importiert die OnlineBestellungen in Mention ###"
4 echo ""
5 echo "—STEP 1 ausführung der get_online_bestellung.php—";
6 $phppfad/php $phppfad_importBestellung/get_online_bestellung.php > $logpfad_php/
  get_online_bestellung.log
7 echo "### Leeren der Tabelle OnlineBestellung auf Nexus-neu ###"
8 $mysqlpfad/mysql -h $mysqlhost -u $mysqluser -p$mysqlpwd -e 'TRUNCATE
  OnlineBestellung' $mysqltestshopdb

```

Quelltext D.28: Shell-Skript zum Import der Daten für die Onlinebestellungen

PHP-Skript für den Import der Daten in das Warenwirtschaftssystem

```

1 <?php
2
3 include("/usr/local/pl-update/bin/Mention2Shop/php/classes.php");
4
5 session_start();
6

```

```

7 include("/usr/local/pl-update/bin/Mention2Shop/php/config.php");
8
9 //entfernt die Soderzeichen aus dem EMail-Text
10 function MailUml($text){
11     $text=str_replace("&","&", $text);
12     $text=str_replace("ä","&#xe4", $text);
13     $text=str_replace("ö","&#xf6", $text);
14     $text=str_replace("ü","&#xfc", $text);
15     $text=str_replace("Ä","&#xc4", $text);
16     $text=str_replace("Ö","&#xd6", $text);
17     $text=str_replace("Ü","&#xdc", $text);
18     $text=str_replace("ß","&#xdf", $text);
19     $text=str_replace(""","&#\"", $text);
20     $text=str_replace(">","&#>", $text);
21     $text=str_replace("<","&#<", $text);
22     return($text);
23 }
24
25 //initialisiert das SESSION-ARRAY um die Belegnummern zu speichern
26 $_SESSION['bestell_arr']=array();
27
28 //ermittelt alle seperaten Bestellungen (eine Bestellung beinhaltet mehrere
    Bestellpositionen(einzelne Artikel))
29 $query_obestkk_kunden = mysql_query("SELECT_distinct( bestellid ),_oskundennr_FROM
    _$onlinebestellungtable", $dbh2);
30
31 //erzeugt ein SESSION-Objekt mit der Kundennummer als Namen des Objektes (fuer
    jeden Kunden ein Objekt welches die Daten fuer OBESTKK enthaelt)
32 while($row = mysql_fetch_array($query_obestkk_kunden)){
33
34     $bestellid=$row['bestellid'];
35     $kundennr=$row['oskundennr'];
36
37     //erstellt ein SESSION-ARRAY um die Bestellnummern zu speichern
38     // somit auch nach der WHILE-Schleife zugriff auf die Session-Objekt
        anhand der im Array gespeicherten Bestellnummer
39     $_SESSION['bestell_arr'][$bestellid] = $bestellid;
40
41     //erstellt das Objekt mit ersten Eigenschaften "BestellID ,Kundennummer"
42     ${"$bestellid"} = new obestkk($bestellid,$kundennr);
43
44     $query_obestkk_daten = mysql_query("SELECT_osbeltyp ,oszeit ,osdatum ,
        oszusatz ,hnr ,aridnr ,opvkpreis ,opstueck ,opvkbrutt ,Name_FROM_
        $onlinebestellungtable_WHERE_ bestellid='".$${"$bestellid"}->
        get_bestellid()."'", $dbh2);
45
46     //eintrag dieser Daten in das jeweilige "KundenObjekt"
47     while($row = mysql_fetch_array($query_obestkk_daten)){
48
49         //eintragen der OBESTKK-Daten in das entsprechende "KundenObjekt
            " Objekt
50         ${"$bestellid"}->add_osbeltyp($row['osbeltyp']);
51         ${"$bestellid"}->add_oszeit($row['oszeit']);
52         ${"$bestellid"}->add_osdatum($row['osdatum']);
53         ${"$bestellid"}->add_oszusatz($row['oszusatz']);
54         ${"$bestellid"}->add_Name($row['Name']);
55
56         //eintragen der OBESTKP-Daten in das entsprechende "KundenObjekt
            " Objekt
57         if($row['aridnr']==$aridnr_versand){
58             ${"$bestellid"}->add_hnr("versand");
59         }
60         elseif($row['aridnr']==$aridnr_nachnahme){
61             ${"$bestellid"}->add_hnr("nachnahme");
62         }
63         else{
64             ${"$bestellid"}->add_hnr($row['hnr']);

```



```

65         }
66         ${"$bestellid"}->add_aridnr($row['aridnr']);
67         ${"$bestellid"}->add_vkpreis($row['opvkpreis']);
68         ${"$bestellid"}->add_stueck($row['opstueck']);
69         ${"$bestellid"}->add_vkbrutt($row['opvkbrutt']);
70     }
71 }
72
73 //solange wie es Bestellungen gibt
74 foreach($_SESSION['bestell_arr'] as $key => $bestellid){
75
76     //leert die Variable gesamt_netto fuer jede Bestellung
77     $gesamt_netto=0;
78
79     //leert die Variable gesamt_brutto fuer jede Bestellung
80     $gesamt_brutto=0;
81
82     //solange wie es Positionen pro Bestellung gibt
83     for($i=0;$i<=${"$bestellid"}->get_last_key();$i++){
84
85         //wenn aridnr_arr keine eintrag hat dann ist es ein Artikel
86         if(${"$bestellid"}->get_aridnr($i)==""){
87
88             //ermittelt aus AEL on die HNR enthalten ist und welche nWert
89             //die Spalte ARSVERKAUF hat (e,E oder leer)
90             $query_hnr_in_ael = mssql_query("SELECT aridnr, arsverkauf FROM
91             $aeltable WHERE arherstnr='".${"$bestellid"}->get_hnr($i)."'
92             ", $dbh1);
93
94             //wenn AEL ein Treffer liefert dann ist die HNR ein
95             //Artikelstamm-Artikel und ARIDNR=OPIDNR
96             if(mssql_num_rows($query_hnr_in_ael)>0){
97
98                 while($row = mssql_fetch_array($query_hnr_in_ael)){
99
100                     //wenn der Artikel (HNR) ein e,E in ARSVERKAUF
101                     //hat also EOL dann Nachricht versenden an
102                     //Betreuer
103                     if((!(strpos($row['arsverkauf'], "e")===false))
104                     ||(!(strpos($row['arsverkauf'], "E")===false))
105                     )){
106                         echo "_____ Position _". $i. " _----\n";
107                         echo "Keine Daten fuer den AEL-Artikel _
108                         _". ${"$bestellid"}->get_hnr($i). " _
109                         _vorhanden!! _Evtl. _EOL!!!\n";
110                         echo "_____ \n";
111
112                         //entfernt die Positions-Daten aus dem
113                         //KundenObjekt
114                         ${"$bestellid"}->remove_and_save_hnr($i)
115                         ;
116                         ${"$bestellid"}->remove_vkpreis($i);
117                         ${"$bestellid"}->remove_stueck($i);
118                         ${"$bestellid"}->remove_vkbrutt($i);
119                     }
120                     else{
121                         //eintragen der OPIDNR in das
122                         //entsprechende "KundenObjekt" Objekt
123                         ${"$bestellid"}->add_opidnr("${
124                         $bestellid"}->get_hnr($i), $row['
125                         aridnr']);
126
127                         //berechnung des GesamtNettoPreises der
128                         //Position
129                         $gesamt_netto_pos=number_format((${"
130                         $bestellid"}->get_vkpreis($i)*{"
131                         "$bestellid"}->get_stueck($i))

```

```

114                                     ,2,".", "");
115                                     //berechnung des GesamtBruttoPreises der
116                                     //Position
117                                     $gesamt_brutto_pos=number_format((${"
118                                     $bestellid"}->get_vkbrutt($i)*${"
119                                     $bestellid"}->get_stueck($i)),2,".",
120                                     "", "");
121                                     //eintragen des GeamtNettoPreises der
122                                     //Position
123                                     ${"$bestellid"}->add_gesamt_netto_pos(
124                                     $i,$gesamt_netto_pos);
125                                     //eintragen des GeamtNettoPreises der
126                                     //Position
127                                     ${"$bestellid"}->add_gesamt_brutto_pos
128                                     ($i,$gesamt_brutto_pos);
129                                     }
130                                     }
131                                     }
132                                     else{ //wenn AEL kein Treffer dann in PRL die prdistrib und
133                                     prbestnr ermitteln
134                                     //schaut in PRL ob es zu der HNR einen oder mehrere
135                                     //Eintraege (prdistrib und prbestnr) gibt
136                                     // wenn mehrere eintraege dann die disties nach wichtung
137                                     //ermitteln
138                                     $query_prl = mssql_query("SELECT count(prdistrib)_AS_
139                                     distri FROM $prltable WHERE prherstnr='".${"
140                                     $bestellid"}->get_hnr($i)."' , $dbh1);
141                                     //wenn 1 distri dann direkt die prdistrib und prbestnr
142                                     //ermitteln und in das KundenObjekt speichern
143                                     if(mssql_result($query_prl,0,0)==1){
144                                     $query_prl_distri_bestnr = mssql_query
145                                     ("SELECT prdistrib , prbestnr FROM
146                                     $prltable WHERE prherstnr='".${"
147                                     $bestellid"}->get_hnr($i)."' ,
148                                     $dbh1);
149                                     //fuegt den prdistrib den KundenObjekt hinzu
150                                     ${"$bestellid"}->add_opdistrib(${"$bestellid"}->
151                                     get_hnr($i), mssql_result(

```

```

152         //eintragen des GeamtNettoPreises der Position
153         ${" $bestellid"}->add_gesamt_brutto_pos($i,
154         $gesamt_brutto_pos);
155     }
156     elseif(mssql_result($query_prl,0,0)>1){ //wenn mehrere
157         Distries vorhanden sind anhand einer Wichtung 1
158         Distri und 1 BestelNr zu ermitteln
159
160         //sammeln aller Distributoren der HNR
161         aus Mention-DB
162         $query_distries = mssql_query("SELECT prdistrib_
163         FROM $prltable WHERE prherstnr='".${"
164         $bestellid"}->get_hnr($i)."'", $dbh1);
165
166         //leert das Array mit den Distributoren fuer
167         eine neuen Artikel
168         $distries_arr=array();
169
170         //fuellt ein Array mit allen Distries des
171         Artikels aus Mention
172         while($row = mssql_fetch_array($query_distries))
173         {
174             $distries_arr[]=$row['prdistrib'];
175         }
176
177         //sammelt alle Distributoren mit Rang aus Nexus-
178         neu.plimport.hierarchie
179         $query_rang = mysql_query("SELECT name, rang FROM
180         $hierarchietable", $dbh3);
181
182         //leert das Array mit den Distributoren und
183         Raengen fuer eine neuen Artikel
184         $distries_rang_arr=array();
185
186         //fuellt ein Array mit allen Distries und Rang
187         while($row = mysql_fetch_array($query_rang)){
188             $distries_rang_arr[$row['name']]=$row['
189             rang'];
190         }
191
192         $rang=mysql_num_rows($query_rang);
193
194         //ermittelt den hoechsten Rang des Distributors
195         aus allen Vorhandenn Distributoren des
196         Artikels
197         //so lange wie Distributoren aus Mention
198         existieren
199         foreach ($distries_arr as $key =>
200             $value) {
201             //wenn der Distributor im Distrie-Rang-Array
202             aus Nexus-neu.plimport.hierarchie
203             existiert
204             if(array_key_exists($value,
205                 $distries_rang_arr)){
206                 //Rang aus Distrie-Rang-
207                 Array wird zu
208                 aktueller Rang
209                 $akt_rang=
210                 $distries_rang_arr[
211                 $value];
212                 //wenn aktueller Rang
213                 KLEINER als Rang
214                 if($akt_rang<$rang){
215                     //aktueller Rang wird zu Rang
216                     $rang=$akt_rang;
217                     $distributor=$value;
218                 }
219             }
220         }

```

```

194         }
195     }
196
197     //fuegt den Distributor den KundenObjekt hinzu
198     ${"$bestellid"}->add_opdistrib("${"$bestellid"}->
        get_hnr($i),$distributor);
199
200     //ermittelt die OPBESTNR anhand der HNR und des
        Distributors
201     $query_opbestnr = mssql_query("SELECT prbestnr
        FROM prtable WHERE prherstnr='.${"$bestellid"}->get_hnr($i)'.
        ' _AND prdistrib='
        . "${"$bestellid"}->get_opdistrib("${"$bestellid"}->get_hnr($i)'.
        ' ', $dbh1);
202
203     //fuegt die BestellNr dem KundenObjekt hinzu
204     ${"$bestellid"}->add_opbestnr("${"$bestellid"}->
        get_hnr($i), mssql_result($query_opbestnr
        ,0,0));
205
206     //berechnung des GesamtNettoPreises der Position
207     $gesamt_netto_pos=number_format((${"$bestellid"}
        )->get_vkpreis($i)*${"$bestellid"}->
        get_stueck($i)),2,".", "");
208
209     //berechnung des GesamtBruttoPreises der
        Position
210     $gesamt_brutto_pos=number_format((${"$bestellid"}->get_vkbrutt($i)*${"$bestellid"}->get_stueck($i)),2,".", "");
211
212     //eintragen des GeamtNettoPreises der Position
213     ${"$bestellid"}->add_gesamt_netto_pos($i,
        $gesamt_netto_pos);
214
215     //eintragen des GeamtNettoPreises der Position
216     ${"$bestellid"}->add_gesamt_brutto_pos($i,
        $gesamt_brutto_pos);
217 }
218 else{ // wenn der Artikel werder in AEL noch in PRL
        vorhanden ist (EOL)
219     echo "_____Position_". $i . "_____\n";
220     echo "Keine_Daten_fuer_den_PRL-Artikel_". ${"$bestellid"}->get_hnr($i) . "_vorhanden!!_Evtl
        . _EOL_!!!\n";
221     echo "_____\\n";
222
223     //entfernt die Positions-Daten aus dem
        KundenObjekt
224     ${"$bestellid"}->remove_and_save_hnr($i);
225     ${"$bestellid"}->remove_vkpreis($i);
226     ${"$bestellid"}->remove_stueck($i);
227     ${"$bestellid"}->remove_vkbrutt($i);
228 }
229 }
230 }
231 else{ //ansonsten ist es eine Versand bzw. Nachnahme position da
        ARIDNR schon vorhanden
232
233     //es handelt sich um Versandkosten HNR=>versand
234     if("${"$bestellid"}->get_aridnr($i)==$aridnr_versand){
235         //eintragen der OPIDNR in das entsprechende "
        KundenObjekt" Objekt
236         ${"$bestellid"}->add_opidnr("versand", "${"$bestellid"}->get_aridnr($i));
237     }
238     else{ //es handelt sich um Versandkosten HNR=>nachnahme

```

```

239         //eintragen der OPIDNR in das entsprechende "
                KundenObjekt" Objekt
240         ${"$bestellid"}->add_opidnr("nachnahme", ${"
                $bestellid"}->get_aridnr($i));
241     }
242
243     //berechnung des GesamtNettoPreises der Position
244     $gesamt_netto_pos=number_format((${"$bestellid"}->get_vkpreis(
        $i)*${"$bestellid"}->get_stueck($i)),2,".", "");
245
246     //berechnung des GesamtBruttoPreises der Position
247     $gesamt_brutto_pos=number_format((${"$bestellid"}->get_vkbrutt
        ($i)*${"$bestellid"}->get_stueck($i)),2,".", "");
248
249     //eintragen des GeamtNettoPreises der Position
250     ${"$bestellid"}->add_gesamt_netto_pos($i,$gesamt_netto_pos);
251
252     //eintragen des GeamtNettoPreises der Position
253     ${"$bestellid"}->add_gesamt_brutto_pos($i,$gesamt_brutto_pos);
254 }
255
256 //Berechnung des GesamtNettoPreises der Bestellung (Addition der
        NettoPreise pro Position)
257 $gesamt_netto+=number_format(${"$bestellid"}->get_gesamt_netto_pos(
        $i),2,".", "");
258
259 //Berechnung des GesamtBruttoPreises der Bestellung (Addition der
        BruttoPreise pro Position)
260 $gesamt_brutto+=number_format(${"$bestellid"}->get_gesamt_brutto_pos
        ($i),2,".", "");
261 }
262
263 //eintragen des GesamtNettoPreises in das KundenObjekt
264 ${"$bestellid"}->add_gesamt_netto(number_format($gesamt_netto,2,".", ""));
265
266 //eintragen des GesamtBruttoPreises in das KundenObjekt
267 ${"$bestellid"}->add_gesamt_brutto(number_format($gesamt_brutto,2,".", ""))
        ;
268
269 //Berechnung des GesamtMwSt der Bestellung aus Brutto-Netto
270 $gesamt_mwst=(${"$bestellid"}->get_gesamt_brutto()-${"$bestellid"}->
        get_gesamt_netto());
271
272 //eintragen des GesamtMWST in das KundenObjekt
273 ${"$bestellid"}->add_gesamt_mwst(number_format($gesamt_mwst,2,".", ""));
274 }
275
276 //solange wie es Bestellungen gibt
277 foreach($_SESSION['bestell_arr'] as $key => $bestellid){
278
279     //neu Sortierung der Arrays fuer den import der Bestellung mit
        Positionen ohne die EOL gegangenenen
280     ${"$bestellid"}->sort_hnr_arr();
281     ${"$bestellid"}->sort_vkpreis_arr();
282     ${"$bestellid"}->sort_stueck_arr();
283     ${"$bestellid"}->sort_vkbrutt_arr();
284
285     //einfuegen der Festen Werte aus der config.php in das KundenObjekt fuer
        OBESTKK
286     ${"$bestellid"}->add_osmankey($osmankey);
287     ${"$bestellid"}->add_oswaehrung($oswaehrung);
288     ${"$bestellid"}->add_osukurs($osukurs);
289     ${"$bestellid"}->add_osueinh($osueinh);
290
291     //wenn es eine Bestellung ist anderen Zusattext als bei Angebot
292     if(strcmp(${"$bestellid"}->get_osbeltyp(),B)==0){
293         ${"$bestellid"}->add_ostext($ostext_bestellung);

```

```

294     }
295     else{
296         ${"$bestellid"}->add_ostext($ostext_angebot);
297     }
298
299     //einfuegen der Fester Werte aus der config.php in das KundenObjekt fuer
        OBESTKP (zu jeder einzelnen Position)
300     ${"$bestellid"}->add_opmankey($opmankey);
301     ${"$bestellid"}->add_opmwst($opmwst);
302     ${"$bestellid"}->add_opwaehrung($opwaehrung);
303     ${"$bestellid"}->add_opukurs($opukurs);
304     ${"$bestellid"}->add_opueinh($opueinh);
305 }
306
307 //solange wie es Bestellungen gibt
308 foreach($_SESSION['bestell_arr'] as $key => $bestellid){
309
310     //auslesen der ZB (Zahlungsbedingung) des Kunden aus Mention DB Kunden
        .KZB001
311     $query_zb = mssql_query("SELECT_KDZB01_FROM_$kundentable_WHERE_KDNUMMER
        ='",$key){$bestellid}->get_kundennummer()."', $dbh1);
312
313     $zb=mssql_result($query_zb,0,0);
314
315     //auslesen der PALFDNR fuer das fuellen in OBESTKK.OSPALFDNR zur Anzeige
        des Ansprechpartners bei uebernahme der Bestellung
316     $query_palfdnr = mssql_query("SELECT_PALFDNR_FROM_$adressspatable_WHERE_
        paname2='",$key){$bestellid}->get_Name()."',_AND_PANUMMER_IN_(SELECT_
        KDADNUMMER_FROM_$kundentable_WHERE_KDNUMMER='",$key){$bestellid"}->
        get_kundennummer()."', $dbh1);
317
318     //nur wenn 1 eindeutiger Ansprechpartner gefunden
319     if(mssql_num_rows($query_palfdnr)==1){
320         $palfdnr=mssql_result($query_palfdnr,0,0);
321     }
322
323     //import der Bestell-Daten in OBESTKK
324     mssql_query("INSERT INTO_$sobestkktable_(OSMANKEY,OSKUNDENNR,OSZB,OSZEIT,
        OSDATUM,OSWAEHRUNG,OSUKURS,OSUEINH,OSUDATUM,OSTEXT,OSBESTDAT,
        OSZUSATZT,OSPALFDNR,OSBRUTTO,OSMWST,OSVKPREIS,OSBELTYP)_VALUES_('$.$
        key){$bestellid"}->get_osmankey()."',',$key){$bestellid"}->
        get_kundennummer()."',',$zb',',$key){$bestellid"}->cut_and_get_oszeit()
        ."',',$key){$bestellid"}->get_osdatum()."',',$key){$bestellid"}->
        get_oswaehrung()."',',$key){$bestellid"}->get_osukurs()."',',$key){$
        bestellid"}->get_osueinh()."',',$key){$bestellid"}->get_osdatum()."
        ',',$key){$bestellid"}->get_ostext()."',',$key){$bestellid"}->
        get_osdatum()."',',$key){$bestellid"}->get_oszusatz()."',',$palfdnr.
        "',',$key){$bestellid"}->get_gesamt_brutto()."',',$key){$bestellid"}->
        get_gesamt_mwst()."',',$key){$bestellid"}->get_gesamt_netto()."',',$key){
        "$bestellid"}->get_osbeltyp()."', $dbh1);
325
326     //ermitteln der MaxLFDNUM aus OBESTKK (muss in OBESTKP eingetragen
        werden und da max() die neueste Bestellung ist)
327     $query_max_bestnr = mssql_query("SELECT_max(OSLFDNUM)_FROM_$sobestkktable
        ", $dbh1);
328
329     //wenn noch kein Bestellung in Mention DB
330     if(mssql_result($query_max_bestnr,0,0)==""){
331         $max_bestnr=1;
332     }
333     else{
334         $max_bestnr=mssql_result($query_max_bestnr,0,0);
335     }
336
337     //solange wie es Positionen pro Bestellung gibt import der Positions-
        Daten in OBESTKP
338     for($i=0;$i<${"$bestellid"}->get_anz_sort_hnr();$i++){

```

```

339 mssql_query("INSERT INTO Sobestkptable (OPMANKEY, OPLFDNUM, OPIDNR
340 , OPVKPREIS, OPSTUECK, OPMWST, OPWAEHRUNG, OPUKURS, OPUEINH,
    OPVKBRUTT, opdistrib, opbestnr) VALUES ('" . ${"$bestellid"}->
    get_opmankey() . "' , '" . $max_bestnr . "' , '" . ${"$bestellid"}->
    get_opidnr(${"$bestellid"}->get_sort_hnr($i)) . "' , '" . ${"$bestellid"}->
    get_sort_vkpreis($i) . "' , '" . ${"$bestellid"}->
    get_sort_stueck($i) . "' , '" . ${"$bestellid"}->get_opmwst() . "' , '" .
    ${"$bestellid"}->get_opwaehrung() . "' , '" . ${"$bestellid"}->
    get_opukurs() . "' , '" . ${"$bestellid"}->get_opueinh() . "' , '" . ${"$bestellid"}->
    get_sort_vkbrutt($i) . "' , '" . ${"$bestellid"}->
    get_opdistrib(${"$bestellid"}->get_sort_hnr($i)) . "' , '" . ${"$bestellid"}->
    get_opbestnr(${"$bestellid"}->get_sort_hnr($i)
    ) . "')" , $dbh1);

341 }
342
343 //wenn mindestens 1 Artikel der Bestellung EOL ist dann EMail an
    Betreuer mit den HNR die EOL sind
344 if(${"$bestellid"}->get_anz_email_arr()>0){
345     //ermitteln der BetreuerID aus Kaufhaus.KundenBetreuer und die
        mit crontab erstellte Kaufhaus.KundenMentionNR
346     $query_persID = mysql_query("SELECT distinct (KB.persID) , KU.Name,
        KU.Vorname, KU.Firma , KU.Strasse , KU.PLZ, KU.Ort , KU.Tel , KU.eMail
        , KU.Fax FROM $kunden_kaufhaus_table as KU,
        $kundenbetreuertable as KB WHERE KU.MentionKundennummer=' " . ${"$bestellid"}->
        get_kundennummer() . "' AND KU.Nummer=KB.kdID"
        , $dbh2);

347
348     $persID=mysql_result($query_persID,0,0);
349     $Name=mysql_result($query_persID,0,1);
350     $Vorname=mysql_result($query_persID,0,2);
351     $Firma=mysql_result($query_persID,0,3);
352     $Strasse=mysql_result($query_persID,0,4);
353     $PLZ=mysql_result($query_persID,0,5);
354     $Ort=mysql_result($query_persID,0,6);
355     $Tel=mysql_result($query_persID,0,7);
356     $eMail=mysql_result($query_persID,0,8);
357     $Fax=mysql_result($query_persID,0,9);
358
359     //ermittelt anhand der perID die EMail-Adresse des Betreuers aus
        Nexus-neu.ProSoft-data.pers
360     $query_mail =mysql_query("SELECT email FROM $perstable WHERE ID
        ='$persID'" , $dbh4);
361
362     $mail=mysql_result($query_mail,0,0);
363
364     // Kopf der Message erzeugen
365     if (${"$bestellid"}->get_osbeltyp()=="A") {
366         $message="Angebot erstellen \n-----\n\n";
367         $Betreff="Online-Shop: ";
368         $Betreff.=" automatisches Online-Angebot";
369     }
370     else {
371         $message="Bestellung \n-----\n\n";
372         $Betreff="Online-Shop: ";
373         $Betreff.=" automatische Online-Bestellung";
374     }
375
376     $message.="Die nachfolgenden Artikel wurden aus dem OnlineShop
        automatisch bestellt (sollten in Mention einfließen) , sind
        jedoch EOL\n\n";
377     $message.="Kundennr. .... : " . ${"$bestellid"}->get_kundennummer() . "
        \n";
378     $message.="An..... : $Name, $Vorname, $Firma\n";
379     $message.="Adresse..... : $Strasse , $PLZ, $Ort\n";
380     $message.="Telefon..... : $Tel\n";
381     $message.="eMail..... : $eMail\n";

```

```

382     $message.="Fax.....:~$Fax\n\n";
383
384     $message.="_____\n";
385     $message.="Nr_HerstNr. .... |~ArtNr.\n";
386     $message.="... Artikelbezeichnung.....Menge....EUR/Stk ... Gesamt\n";
387     $message.="_____\n\n";
388
389     $l=1; //Anzeige wieviele Artikel
390     $klammer=")"; //optik für Ausgabe
391     $trenn="|"; //optik für Ausgabe
392     $leer=""; //optik für Ausgabe
393
394     //solange wie es EOL-Artikel gibt EMail-Text bauen
395     for ($i=0;$i<${"$bestellid"}->get_anz_email_arr();$i++){
396
397         if(${"$bestellid"}->get_opidnr("${"$bestellid"}->
398             get_sort_hnr($i))!=""){
399             $message.=sprintf("%ld%ls%ls%ls%4s%6s\n", (int)
400                 $l, (string)$klammer, (string)$leer, (string){
401                     "$bestellid"}->get_sort_hnr($i), (string)
402                     $trenn, (int){"$bestellid"}->get_opidnr("${
403                         $bestellid"}->get_email_hnr($i)));
404             }
405             else{
406                 $message.=sprintf("%ld%ls%ls%ls\n", (int)$l, (
407                     string)$klammer, (string)$leer, (string){"$
408                         bestellid"}->get_email_hnr($i));
409             }
410             $message.=sprintf("%3s%-4ls%5s%13.2f%9.2f\n", (string)
411                 $leer, (string)$ben, (string)str_replace(".000", "", ${
412                     "$bestellid"}->get_email_stueck($i)), (double)({"$
413                     bestellid"}->get_email_preis_netto($i)), (double)({"$
414                     bestellid"}->get_email_preis_netto($i)*${"$
415                     bestellid"}->get_email_stueck($i)));
416             $message.="~\n";
417
418             $l++;
419         }
420     }
421
422     $message.="~\n\n Alle Preise sind Netto-Preise zuzueglich 19%
423     MwSt!!!! ";
424
425     // Umlaute zurueckverwandeln
426     $message=MailUml($message);
427
428     //versendet die E-Mail
429     Mail($mail, $Betreff, $message, "From:~". $Vorname. "~". $Name. "<".
430         $eMail. ">");
431 }
432 ?>

```

Quelltext D.29: PHP-Skript für den Import der Daten in das Warenwirtschaftssystem

Glossar

Array

Ein Array ist eine Datenstruktur in der Informatik. Es ermöglicht das geordnete Speichern von mehreren Variablen eines Datentyps. Auf diese Variablen kann über einen Index wieder zugegriffen werden [19].

Client

Als Client wird ein Computerprogramm bezeichnet, das nach dem Client-Server-Modell eine Verbindung mit einem Server aufnimmt und Nachrichten mit diesem austauscht [21].

Cookie

Ein Cookie ist ein kleiner Eintrag im Dateiverzeichnis des Computers. Darin sind zeitlich begrenzte Informationen enthalten, die von Webanwendungen aufgerufen und verarbeitet werden können [22].

cron

Der *cron*-Daemon ist eine Jobsteuerung von UNIX, die wiederkehrende Aufgaben automatisch zu einer bestimmten Zeit ausführen kann [17].

CSV

Ist ein Standard für eine Textdatei zur Speicherung und Austausch von Daten. Die einzelnen Daten werden in häufigen Fällen durch ein Semikolon voneinander getrennt [20].

diff

diff ist ein UNIX-Programm, mit dem zwei Textdateien miteinander verglichen werden können [23].

EK-Preis

Unter EK-Preis versteht man den Einkaufspreis eines Produktes bei einem Hersteller oder einem Zulieferer.

EOL

EOL bedeutet End of Life. Ist ein Artikel damit gekennzeichnet, wird dieser nicht mehr produziert und ist demnächst nicht mehr am Markt verfügbar.

Extranet

Ein Extranet ist eine Erweiterung des Intranets, die nur von einer festgelegten Gruppe externer Benutzer verwendet werden kann. Es ist nur durch eine separate Standleitung oder Einwahl über das Internet erreichbar [24].

Flag

Mit Flag wird eine binäre Variable oder ein Statusindikator bezeichnet, welcher als Hilfsmittel zur Kennzeichnung bestimmter Zustände benutzt werden kann [25].

FTP

FTP ist ein Netzwerkprotokoll zur Übertragung von Dateien über TCP/IP-Netzwerke [26].

GET

Die GET-Variable ist ein Bestandteil der Argumentenübertragung bei HTTP. Die Werte einer Variable werden dabei durch die URL übertragen [27].

GNU

Ist die abgekürzte Bezeichnung für ein Projekt zur Entwicklung eines vollständig freien Betriebssystems. GNU kann mit GNU is not UNIX übersetzt werden [28].

HTML

HTML ist eine textbasierte Auszeichnungssprache zur Strukturierung von Inhalten wie Texten, Bildern und Hyperlinks in HTML-Dokumenten [29].

HTTP

HTTP ist ein Protokoll zur Übertragung von Daten über ein Netzwerk. Es wird hauptsächlich eingesetzt, um Webseiten aus dem World Wide Web an einem Webbrowser zu übertragen [27].

ID

ID ist die Kurzform von identifier. Diese Kurzform wird häufig in der Informatik als Variablenname, zum Beispiel User-ID oder Session-ID, verwendet.

Intranet

Ein Intranet ist ein organisations- oder unternehmensinternes, nichtöffentliches Rechnernetzwerk. Es steht Mitarbeitern einer Unternehmung oder Organisation als Informations-, Kommunikations- und Anwendungsplattform zur Verfügung [30].

LAMP

LAMP ist ein Akronym für den kombinierten Einsatz von Programmen auf Basis von Linux, um dynamische Webseiten zur Verfügung zu stellen [31].

Link

Als Link bezeichnet man einen Querverweis auf ein anderes Dokument innerhalb des World Wide Web. [32]

Master

Als Master, zu deutsch Herr, wird in dieser Arbeit der Teilnehmer bezeichnet, der bei dem Datenabgleich die zu ändernden Daten vorgibt.

PHP

PHP ist eine Skriptsprache mit einer an C angelehnten Syntax, die hauptsächlich zur Erstellung von dynamischen Webseiten oder Webanwendungen verwendet wird [33].

Pop-Up

Ein Pop-Up ist ein visuelles Element bei Webanwendungen, um Informationen über dem eigentlichen Inhalt der Seite darzustellen [34].

POST

Die POST-Variable ist ein Bestandteil der Argumentenübertragung bei HTTP. Die Werte einer Variable werden dabei versteckt und nicht durch die URL übertragen [27].

Request

Ein Request, zu deutsch Anfrage, ist Anforderung eines Clients an den Server, dessen Dienst er benötigt [35].

Response

Ein Response, ist die Antwort des Servers auf eine Anforderung eines Clients [35].

scp

scp ist die Abkürzung für Secure Copy Protokoll. Secure Copy Protokoll ist ein Protokoll sowie ein Programm zur verschlüsselten Übertragung von Daten zwischen zwei Computern über ein Rechnernetz [37.]

Select

Ein SELECT wird zum Abrufen von Datensätzen aus einer Datenbank verwendet.

Server

Ein Server ist ein Programm, das einen Dienst anbietet. Im Rahmen des Client-Server-Modells kann ein anderes Programm, der Client, diesen Dienst nutzen. Clients und Server können als Programme auf verschiedenen Rechnern oder auf demselben Rechner laufen [36].

Slave

Als Slave, zu deutsch Sklave, wird in dieser Arbeit der Teilnehmer bezeichnet, der bei dem Datenabgleich die zu ändernden Daten erhält und diese ändert.

SSH

SSH steht für Secure Shell und ist ein auf Client/Server-Architektur basierendes Protokoll, das einem Client eine sichere Verbindung zu einem entfernten Server ermöglicht [38].

URL

URL steht für Uniform Resource Locator. Mit einer URL wird der Ort einer Ressource bei Verwendung der Protokolle HTTP und FTP eindeutig bestimmt.

VK-Preis

Unter VK-Preis versteht man den Verkaufspreis eines Produktes an einen Kunden.

VPN

Virtual Private Network dient der Einbindung von Geräten eines benachbarten Netzes an das eigene Netz, ohne dass die Netzwerke zueinander kompatibel sein müssen [39].

World Wide Web

Das World Wide Web ist ein über das Internet abrufbares Hypertext-System. Hierzu benötigt man einen Webbrowser, um die Daten vom Webserver zu holen und z. B. auf dem Bildschirm anzuzeigen. Der Benutzer kann den Hyperlinks im Dokument folgen, die auf andere Dokumente verweisen, gleichgültig ob sie auf demselben Webserver oder einem anderen gespeichert sind. Hierdurch ergibt sich ein weltweites Netz aus Webseiten [40].

XML

Die Extensible Markup Language ist eine Metasprache zur Beschreibung von Sprachen. Damit werden Daten und deren Struktur in einem unabhängigen Format beschrieben und gespeichert. [41] S. 11 & S. 12

Literaturverzeichnis

- [1] Stammdaten
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/Stammdaten>

- [2] Distributor
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/Distributor>

- [3] Aufbau von PHP
verfügbar am: 13.02.2009
<http://www.mathematik.uni-ulm.de/sai/ws01/portalsem/rene/internetportale/node11.html>

- [4] Was ist PHP
verfügbar am: 13.02.2009
<http://www.teialehrbuch.de/Kostenlose-Kurse/PHP/9207-Was-ist-PHP.html>

- [5] Session-Management
verfügbar am: 13.02.2009
<http://www.mathematik.uni-ulm.de/sai/ws01/portalsem/rene/internetportale/node15.html>

- [6] Objektorientierte Programmierung
verfügbar am: 13.02.2009
http://de.wikipedia.org/wiki/Objektorientierte_Programmierung

- [7] Klassendiagramm
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/Klassendiagramm>

- [8] UML
verfügbar am: 13.02.2009

<http://ivs.cs.uni-magdeburg.de/~dumke/UML/2.htm>

[9] Dynamische Webseiten

verfügbar am: 13.02.20097

<http://www.nomios.de/leistungen/diverse-artikel/dynamische-webseiten/>

[10] Ablauf einer PHP-Anfrage

verfügbar am: 13.02.2009

<http://www.mathematik.uni-ulm.de/sai/ws01/portalsem/rene/internetportale/node10.html>

[11] PHPLIB

verfügbar am: 13.02.2009

<http://kris.koehntopp.de/artikel/phplib-deutsch/>

[12] Was ist MySQL

verfügbar am: 13.02.2009

<http://dev.mysql.com/doc/refman/5.1/de/what-is.html>

[13] phpMyAdmin

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/PhpMyAdmin>

[14] Was ist UNIX

verfügbar am: 13.02.2009

<http://www.netzmafia.de/skripten/unix/unix1.html#1.1>

[15] Geschichte von UNIX

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Unix-Derivat>

[16] Debian

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Debian>

[17] cron-Daemon

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Cron>

[18] *cronjobs* mit *crontab*

verfügbar am: 13.02.2009

<http://www.linuxhilfen.org/befehle/crontab.html>

[19] Array

verfügbar am: 13.02.2009

[http://de.wikipedia.org/wiki/Feld_\(Datentyp\)](http://de.wikipedia.org/wiki/Feld_(Datentyp))

[20] CSV

verfügbar am: 13.02.2009

[http://de.wikipedia.org/wiki/CSV_\(Dateiformat\)](http://de.wikipedia.org/wiki/CSV_(Dateiformat))

[21] Client

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Client>

[22] Cookie

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Cookie>

[23] Diff

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Diff>

[24] Extranet

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/Extranet>

[25] Flag

verfügbar am: 13.02.2009

[http://de.wikipedia.org/wiki/Flag_\(Informatik\)](http://de.wikipedia.org/wiki/Flag_(Informatik))

[26] FTP

verfügbar am: 13.02.2009

http://de.wikipedia.org/wiki/File_Transfer_Protocol

[27] HTTP

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/HTTP>

[28] GNU

verfügbar am: 13.02.2009

<http://de.wikipedia.org/wiki/GNU-Projekt>

- [29] HTML
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/HTML>
- [30] Intranet
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/Intranet>
- [31] LAMP
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/LAMP>
- [32] Link
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/Hyperlink>
- [33] PHP
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/PHP>
- [34] Pop-Up
verfügbar am: 13.02.2009
<http://de.wikipedia.org/wiki/Pop-up>
- [35] Request und Response
verfügbar am: 17.02.2009
<http://de.wikipedia.org/wiki/Client-Server-Modell>
- [36] Server
verfügbar am: 17.02.2009
<http://de.wikipedia.org/wiki/Server>
- [37] Secure Copy
verfügbar am: 17.02.2009
http://de.wikipedia.org/wiki/Secure_Copy
- [38] Secure Shell
verfügbar am: 17.02.2009
http://www.securitymanager.de/magazin/artikel_1459-print_hacking_ssh_secure_shell.html

[39] VPN

verfügbar am: 17.02.2009

http://de.wikipedia.org/wiki/Virtual_Private_Network

[40] World Wide Web

verfügbar am: 17.02.2009

http://de.wikipedia.org/wiki/World_Wide_Web

[41] XML

Erlenkötter, Helmut: Extensible Markup Language von Anfang an. - 1. Aufl. - Reinbek bei Hamburg: Rowohlt Taschenbuch Verlag GmbH, September 2003

Eidesstattliche Erklärung

Hiermit versichere ich eidesstattlich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter der Quelle kenntlich gemacht.

Mittweida, den 13. März 2009

(Sebastian Eichelbaum)